AFRL-RQ-WP-TR-2014-0191

INTEGRATED HUMAN BEHAVIOR MODELING AND STOCHASTIC CONTROL (IHBMSC)

Phillip R. Chandler

Autonomous Control Branch
Power and Control Division

Michael J. Patzek

Supervisory Control and Cognition Branch
Warfighter Interface Division

Meir Pachter

Air Force Institute of Technology

Clayton Rothwell, Sara Naderer, and Krishna Kalyanam

InfoSciTex Corporation

AUGUST 2014
Final Report

Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

STINFO COPY

AIR FORCE RESEARCH LABORATORY
AEROSPACE SYSTEMS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7541
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE AND SIGNATURE PAGE

AFRL-RQ-WP-TR-2014-0191 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//

PHILLIP R. CHANDLER
Project Manager
Control Automation Branch
Power and Control Division

//Signature//

JEFFREY C. TROMP, Chief
Control Automation Branch
Power and Control Division

//Signature//

DANIEL B. THOMPSON
Principal Scientist
Power and Control Division
Aerospace Systems Directorate

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YY)***
August 2014

2. REPORT TYPE
Final

3. DATES COVERED ***(From - To)***
01 January 2013 – 15 August 2014

4. TITLE AND SUBTITLE
INTEGRATED HUMAN BEHAVIOR MODELING AND STOCHASTIC CONTROL (IHBMSC)

5a. CONTRACT NUMBER
In-house

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62201F

6. AUTHOR(S)

Phillip R. Chandler (AFRL/RQQA)
Michael J. Patzek (AFRL/RHCI)
Meir Pachter (AFIT)
Clayton Rothwell, Sara Naderer, and Krishna Kalyanam (InfoSciTex Corporation)

5d. PROJECT NUMBER
2403

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
Q05C

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Autonomous Control Branch (AFRL/RQQA)
Power and Control Division
Air Force Research Laboratory, Aerospace Systems Directorate
Wright-Patterson Air Force Base, OH 45433-7541
Air Force Materiel Command, United States Air Force

Supervisory Control and Cognition Branch (AFRL/RHCI)
Warfighter Interface Division, Air Force Research Laboratory
711 Human Performance Wing
Wright-Patterson Air Force Base, OH 45433
Air Force Materiel Command, United States Air Force

Air Force Institute of Technology (AFIT)
2950 Hobson Way
Wright-Patterson Air Force Base, OH 45433

InfoSciTex Corporation
4027 Colonel Glenn Highway
Suite 210
Dayton, OH 45431-1672

8. PERFORMING ORGANIZATION REPORT NUMBER
AFRL-RQ-WP-TR-2014-0191

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Aerospace Systems Directorate
Wright-Patterson Air Force Base, OH 45433-7541
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RQQA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RQ-WP-TR-2014-0191

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
PA Case Number: 88ABW-2014-4334; Clearance Date: 15 Sep 2014.

14. ABSTRACT

This project developed an operator error model for improving the accuracy of the operator task of discriminating objects in multiple UAV videos on the Vigilant Spirit Control Station (VSCS) ground station. This error model is idiographic in the sense that it is tailored for a specific individual. The operator error model is used in a stochastic controller which decides, based on the model uncertainty and the operator input of target/non-target, whether to revisit the object to obtain additional information or to continue to the next object. The stochastic controller computes the probability of a target given the available evidence and the reliability of the evidence as per the operator error model. Over 120 operator-in-the-loop tests were conducted to obtain sufficient data to be able to estimate the false alarm (FA) and missed detection (MD) rates over four variables (altitude, aspect angle, dwell time, workload). For a 50% target density scenario where the operator discrimination task is fairly difficult, the integrated human behavior model and stochastic controller (IHBMSC) was shown to reduce the FA rate from 35% to 5%. Also, the IHBMSC performance was found to be quite robust for lower target densities (5%) and different operators.

15. SUBJECT TERMS

human behavior modeling, idiographic, operator error modeling, stochastic control, confusion matrix, signal detection theory, SDT, information gain, IG, operator cueing, false alarm, missed detection

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE
Unclassified	Unclassified	Unclassified

17. LIMITATION OF ABSTRACT:
SAR

18. NUMBER OF PAGES
190

19a. NAME OF RESPONSIBLE PERSON (Monitor)
Phillip R. Chandler

19b. TELEPHONE NUMBER *(Include Area Code)*
N/A

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

List of Symbols

Symbol	Definition
P(T)	probability of target
P_{TR}	probability of target report
P_{FTR}	probability of false target report
P(T|Op=T)	probability of target given operator said target
P(Op=T|T)	probability operator said target given target

List of Acronyms

Acronym	Definition
AA	Aspect Angle
AGL	Above Ground Level
ATR	Automatic Target Recognition
CCA	Cooperative Control Algorithms
DT	Dwell Time
FA	False Alarm
GSD	Ground Sample Distance
IET	Inter-Event Time
IG	Information Gain
IHBMSC	Integrated Human Behavior Modeling and Stochastic Control
LEAP	Learning Estimates of Aggregate Performance
MD	Missed Detection
MSL	Mean Sea Level
PC	Percent Correct
PST	Point Search Task
POI	Point of Interest
POIs	Points of Interest
ROC	Receiver Operating Characteristic
RPG	Rifle Propelled Grenade
RSTA	Reconnaissance, Surveillance and Target Acquisition
SC	Stochastic Controller
SDT	Signal Detection Theory
std	Standard Deviation

Acronym	Definition
TSD	Tactical Situation Display
UAV	Unmanned Aerial Vehicle
VSCS	Vigilant Spirit Control Station

INTEGRATED HUMAN BEHAVIOR MODELING
AND
STOCHASTIC CONTROL

I. Introduction

Given advances in automation, systems are now being considered with the concept of a single operator supervising multiple Unmanned Aerial Vehicles (UAVs) performing cooperative missions. Recent cooperative control technology efforts [2, 5] have focused on developing and demonstrating multi-UAV assignment, scheduling and route planning for Reconnaissance, Surveillance and Target Acquisition (RSTA) tasks. These problem-solving capabilities can produce optimal or near-optimal solutions to minimize mission time or total distance traveled by a team of UAVs as they visit designated target locations. These capabilities have been shown to reduce operator workload and the time required to generate solutions compared to performing these tasks manually. The human-automation interaction, however, is fairly rigid, limiting flexibility in mission planning and execution. The operator identifies target locations to visit, selects the number of UAVs to be assigned, and the cooperative control automation solves the problem and presents the solution for the operator to accept or reject. It is more or less an "all or none" proposition, with little ability to make adjustments without adversely affecting the solution's optimality. An in-house effort is underway to expand beyond the "management by consent" mode of interaction by developing adaptable automation and interaction methods, where the operator will have options (e.g., set duration, hold on target, skip target, change target ordering, etc.) available to adjust or fine tune the cooperative control problem-solving process and solution to satisfy both the mission and operator needs. The goal is to harness the efficiencies afforded from the computer algorithms while providing more flexible control to the operator. With *adaptable* automation and interaction methods under development, the next step is to facilitate *adaptive* interfaces and automation where the machine initiates minor to major changes to the operator controls, displays and mission execution. Here this adaptive capability will be achieved through mutual understanding, where the machine automation has some awareness of the individual, and the human has in-depth knowledge about the automation as ‘they’ perform the mission.

In pursuit of adaptive interface and automation methods, the objective of this research is to develop a reliable human behavioral model for inclusion in a stochastic controller which will be coupled to the vehicle assignment, scheduling and path planning automation. The goal is to enable the automation to perform both a priori and real-time mission planning and problem-solving based on the mission situation, vehicle resources and the individual’s capacity to effectively perform the associated work. Moving away from the “one size fits all” philosophy, the effort is developing a human behavioral model that represents an individual’s performance and biases across a range of target acquisition and workload situations (e.g., target arrival rate, dwell time over potential targets, etc.). The idiographic-based operator model is then integrated into a stochastic controller and cooperative control algorithm, leading to a more “closed-loop” form of problem-solving that not only accounts for the vehicles’ capabilities but the individual operator’s behavior and performance as well. The resulting tailored mission decisions and plans are expected to increase the system’s target acquisition performance, while reducing performance variability across users.

The research approach leverages and extends current developments in stochastic control optimization [6, 7] by introducing a representation of an individual’s behavior to the stochastic controller. The goal of the embedded stochastic controller is to plan vehicle assignments, schedules and routing to reduce the system’s probability of false alarms without adversely affecting the target detections using a priori and real-time situation and performance data. Presently, the stochastic controller analyzes the operator's response and computes whether or not the object should be revisited by referencing target acquisition probabilities, as well as the remaining objects to be inspected and the resources available. The basis of the current stochastic control automation is aircraft/sensor position and time factors relative to the target(s) as these variables had significant bearing on target acquisition performance during recent human-in-the-loop simulations. AFRL/RQQA's initial stochastic controller design for sequential inspection of objects explored the decision of whether or not a UAV should revisit the object of interest to maximize the total expected information gain given there are multiple objects to visit, limited resources, a set of corresponding conditions (position, time, and workload) to factor in, and the operator's target assessment.

In human-in-the-loop experiments that helped identify the initial set of stochastic control parameters, and in other experiments with related multi-UAV tasks involving visual inspection of targets, a wide variance in performance was observed across operators. Given this, a human behavioral model that is based on the group's average performance may not represent a significant portion of the users very well, resulting in mismatched plans and UAV actions for a number of end-users. The research reported here takes a different approach by representing the individual's behavior and performance in a computational human behavioral model (also referred to as the "operator error model" in this report) which, in essence, will customize the problem-solving and resulting mission plans for the individual operator at the control station. During the mission, the controller can be used to monitor the actual states (e.g., ground-sampled distance, aspect angle, and dwell time) and compare it to the expected states, computing the probability of detecting a given target. When disparities exist between what was reported versus expected, or where conditions are beyond prescribed levels, the system can adapt mission plans and actions, manipulating such things as the UAVs' routes, automated sensor steering action points, and the employment of video inspection tools (e.g., digital video recording methods, mosaics).

1.1 The Stochastic Controller

1.1.1 Overview.

This section describes the formulation of the stochastic control algorithm that aides the operator in the sequential inspection task. The logic behind the controller is the following: on the first look at an object, the operator is unaided by the controller and makes a call as to whether or not the object is a target. Since the state corresponding to the 1^{st} look, $s = (d, a, t, i)$ [GSD,AA,DT,IET] is random and is possibly a bad state, the controller is designed to determine an "optimal" (to be precisely defined soon) re-visit state. Note that the controller may also decide not to re-visit the object. So as to determine the optimal re-visit state, the controller takes into account all of the following: the state for the 1^{st} look, the operator's response, the number of re-visits left and the likelihood of the object being a target (or non-target), which is modeled by the a priori target density. The controller design is tuned to a particular operator model, which is specified by the probabilities of true and false target reports, i.e., $P_{TR}(s)$ and $P_{FTR}(s)$, both of which are functions of the state s—we refer to the operator's confusion matrix, which is state dependent.

Operator Error Model

The operator is treated as a sensor-in-the-loop automaton. The operator is not infallible and this is statistically accounted for by his confusion matrix in the design of the stochastic controller. To quantify the operator's performance, two random variables are considered: the variable X that specifies whether the object is the target T or false target FT and the operator decision, Z which specifies whether he determines the object to be a target i.e., $Z = Z_1$ or a false target, $Z = Z_2$. Let the a priori probability that an object is a target a.k.a target density, $Prob(X = T) = p$. The conditional probabilities which specify whether the operator correctly reported a target or a false target are,

$$P_{TR}(s) := Prob\{Z = Z_1 | X = T, s\} \text{ and} \tag{1.1}$$

$$P_{FTR}(s) := Prob\{Z = Z_2 | X = FT, s\}. \tag{1.2}$$

Together, $P_{TR}(s)$ and $P_{FTR}(s)$ determine the entries of the binary "confusion matrix" of the operator for a given state, s. The mutual information between the random variables X and Z, derived along the lines of information theory [15], is:

$$\begin{aligned} \mathcal{I}(X;Z) &= H(X) - H(X|Z) \\ &= \sum_{x,z} Prob\{X = x, Z = z\} \log \frac{Prob\{X = x, Z = z\}}{Prob\{X = x\}Prob\{Z = z\}} \end{aligned} \tag{1.3}$$

where $H(X)$ is the entropy of the random variable X and $H(X|Z)$ is the conditional entropy of X given the observation Z. Using Bayes' rule and the probabilities (1.1) and (1.2), one can show that the mutual information takes the form:

$$\begin{aligned} \mathcal{I}(s,p) = \quad & pP_{TR}(s) \log \frac{P_{TR}(s)}{pP_{TR}(s) + (1-p)(1-P_{FTR}(s))} + \\ & p(1-P_{TR}(s)) \log \frac{1-P_{TR}(s)}{p(1-P_{TR}(s)) + (1-p)P_{FTR}(s)} + \\ & (1-p)(1-P_{FTR}(s)) \log \frac{1-P_{FTR}(s)}{pP_{TR}(s) + (1-p)(1-P_{FTR}(s))} + \\ & (1-p)P_{FTR}(s) \log \frac{P_{FTR}(s)}{p(1-P_{TR}(s)) + (1-p)P_{FTR}(s)}. \end{aligned} \tag{1.4}$$

We have the probabilities that the operator will determine the object to be a target or a false target, after the 1^{st} look, given by

$$q_1(s) := Prob\{Z = Z_1\} = pP_{TR}(s) + (1-p)(1-P_{FTR}(s)), \tag{1.5}$$

$$q_2(s) := Prob\{Z = Z_2\} = p(1-P_{TR}(s)) + (1-p)P_{FTR}(s), \tag{1.6}$$

respectively. Suppose after the 1^{st} look, the operator determines the object to be a target. Using Bayes' rule, we have the a posteriori probability that the object is indeed a target given by,

$$p^+(s, Z_1) := Prob\{X = T | Z = Z_1\} = \frac{pP_{TR}(s)}{pP_{TR}(s) + (1-p)(1-P_{FTR}(s))} \tag{1.7}$$

On the other hand, if the operator determines the object to be a false target, then

$$p^+(s, Z_2) := Prob\{X = T|Z = Z_2\} = \frac{p(1 - P_{TR}(s))}{p(1 - P_{TR}(s)) + (1 - p)P_{FTR}(s)} \tag{1.8}$$

1.1.2 Sequential Inspection.

The Stochastic Controller is designed to control a sequential inspection scenario. The sequential inspection problem has a long history and is often called “The Secretary Problem” or the “Parking Problem”. The secretary problem is stated as an interviewer choosing the “best” secretary from a sequence of *n* candidates, where, when an individual is interviewed, the secretary can be hired, or if not hired, can no longer be considered. The decision [Hire, Not-Hire] is final. Hence for the Target inspection problem, if the decision is made to not revisit the object, then the UAV must continue to the next object in the sequence—there is no going back later. This significantly simplifies the complexity of the problem. The solution is generally implemented as a Dynamic Program with just *n* stages with no transitions from, say stage *n* to $n-1, n-2...1$. The optimal solution for the secretary problem is to interview 1/3 of the candidates, grade them from 1-10, note the highest grade so far, then hire the next candidate in the sequence that has a grade ≥ the high grade from the first third.

Instead of selecting just one secretary, in the SC problem there can be *m* revisits, where $m < n$. For the scenarios in this report, $n = 5$ and $m = 3$. The number of revisits allowed is part of the problem specification. In the secretary problem there is no information about the candidate population. In essence, there is an estimation problem in trying to approximate, *during the decision process*, what the average grade and variation is. In contrast, the SC has already modeled the variability in the process before the objects are observed. For example, $P(T) = p$ or the prior probability that an object is a Target is assumed known. Also, the errors in the operator observation is modeled as a confusion matrix. This allows the SC to answer the question: what is the expected value of revisiting object *i* now, versus the expected value of using the revisit on one of the remaining objects?

The equivalent question cannot be addressed in the secretary problem: What is the expectation that a higher score will be seen in the remaining candidates? The information needed to compute this expectation comes from the first third of the candidates. There is no guarantee that this approach will yield the top scoring candidate, but will, on average, yield a high score. The sequential nature of the information limits the achievable performance. The longer the sequence (the better the estimate of the distribution), the greater the probability of the top scorer being selected. The SC, in contrast, has a model of the error distribution up front. If there is no modeling error, and the expectation is accurate, then the achievable performance should be maximum [9].

1.1.3 Objective Function.

The objective is to aid the operator in classifying the object correctly as a target or a non-target. Let the state of the system, $s = (d, a, t, i)$, $s \in \mathcal{S}$, where $\mathcal{S}$ is the set of all possible states. These constitute all possible combinations of the independent variables: Ground Sample Distance (GSD), Aspect Angle (AA), Dwell Time (DT) and Inter-Event Time (IET) i.e., $d \in \mathcal{D}, a \in \mathcal{A}, t \in \mathcal{T}, i \in \mathcal{E}$. So, the set of all states, $\mathcal{S} = \mathcal{D} \times \mathcal{A} \times \mathcal{T} \times \mathcal{E}$. Suppose after the 1^{st} look in state s_1, the operator's decision is Z. We wish to give the operator a second look, in an attempt to show the object in a "better" state. The question we pose is the following: given the operator error model ($P_{TR}(s)$ and $P_{FTR}(s)$ values $\forall s \in \mathcal{S}$), what state is the optimal one for a second look? Here, the controlled variables are GSD and AA. The other two are assumed to be random variables with known uniform distributions. To answer this question, we solve the following information maximization problem:

$$\max_{d \in \mathcal{D}, a \in \mathcal{A}} \mathbf{E}_{t,i}\{\mathcal{I}(s_2, p^+(s_1, Z))\}, \;\; s_2 = (d, a, t, i), \tag{1.9}$$

where we optimize over the distance, d and angle a and take expectation over the random variables, t and i. We assume a uniform distribution for the random variables. Let

$\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$ and $\mathcal{E} = \{i_1, \ldots, i_{|\mathcal{E}|}\}$. So, the expectation is calculated as follows:

$$\mathbf{E}_{t,i}\{\mathcal{I}(s_2, p^+(s_1, Z))\} = \frac{1}{|\mathcal{T}||\mathcal{E}|} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{E}} \mathcal{I}(s_2, p^+(s_1, Z)), \; s_2 = (d, a, t, i), \tag{1.10}$$

In (1.9), the a posteriori probability, $p^+(s_1, Z)$ serves as the prior probability for the 2^{nd} look and is computed according to (1.7) or (1.8) depending on the operator's decision Z after the 1^{st} look. Let the maximizing ground scale distance and aspect angle be given by $d^*(s_1, Z)$ and $a^*(s_1, Z)$ respectively. So, if a 2^{nd} look was available, the operator would be presented with a state corresponding to distance and angle given by $d^*(s_1, Z)$ and $a^*(s_1, Z)$ respectively. Let the corresponding maximal information be given by $V^*(s_1, z)$.

Maximal information - Look up table

For all states, $s \in \mathcal{S}$ and operator inputs, $Z \in \{Z_1, Z_2\}$, we pre-compute the maximal information, $V^*(s, Z)$ and corresponding optimal re-visit distance and angle, $d^*(s, Z)$ and $a^*(s, Z)$ by solving (1.9). Due to fuel and other constraints, it is possible that one may not have the luxury of having a 2^{nd} look for every object. In general, the operator is tasked with classifying N objects, with the option of having R ($< N$) revisits, i.e., only R out of the N objects can be looked at a second time. So, one has to make a judicious choice as to which objects are worth having a 2^{nd} look at. Again, we use the notion of maximizing information to make our selection.

Suppose we have two objects to look at, i.e., $N = 2$ and $R = 1$. After looking at the 1^{st} object at state s_1, let the operator's decision be Z. Given that we have only one revisit (left), should the operator utilize it for the 1^{st} object or should he keep it in reserve for the 2^{nd} object? To answer this question, we compare the likely information (gain) between the two scenarios:

$$\begin{aligned} &\max\{V^*(s_1, Z), \mathbf{E}_s \mathbf{E}_{\bar{Z}} V^*(s, \bar{Z})\} \\ &= \max\{V^*(s_1, Z), \mathbf{E}_s \left[q_1(s) V^*(s, Z_1) + q_2(s) V^*(s, Z_2)\right]\}. \end{aligned} \tag{1.11}$$

Let the control decision (whether or not to re-visit) be represented by $u^*(k, r, s_1, Z) \in \{0, 1\}$, for the k^{th} object in the sequence of N objects, where r is the number of re-visits left. Here, $u^*(k, r, s_1, Z) = 1$ implies that the decision is made to re-visit the k^{th} object. Let the information corresponding to the optimal decision be, $W^*(k, r, s, Z)$. For the $N = 2$ and $R = 1$ scenario, we have

$$W^*(1, 1, s_1, Z) \quad = \quad \max\{V^*(s_1, Z), \mathbf{E}_s \left[q_1(s)V^*(s, Z_1) + q_2(s)V^*(s, Z_2)\right] \qquad (1.12)$$

The logic behind (1.12) is the following. If a re-visit is utilized at the 1^{st} object, we get the immediate payoff, $V^*(s_1, Z)$. On the other hand, if the re-visit is not utilized, then we have 1 re-visit left at the 2^{nd} object and the corresponding expected return is the 2^{nd} term of the maximization in (1.12). Note that the expectation is over two random variables. First, we take an expectation over the operator's decision and then we take an expectation over the 1^{st} look state for the 2^{nd} object. As before (1.10), we assume uniform distribution of the state and compute the expectation as follows:

$$\mathbf{E}_s f(s) = \frac{1}{|\mathcal{D}||\mathcal{A}||\mathcal{T}||\mathcal{E}|} \sum_{d\in\mathcal{D}} \sum_{a\in\mathcal{A}} \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{E}} f(s). \qquad (1.13)$$

The optimal control corresponding to (1.12) is given by,

$$u^*(1, 1, s_1, Z) = 1, \text{ if } W^*(1, 1, s_1, Z) = V^*(s_1, Z), \qquad (1.14)$$

and 0 otherwise.

1.1.4 Dynamic Programming Algorithm.

We extend the control algorithm to the general case of N (≥ 2) objects and R ($< N$) revisits. To do so, we employ a backward recursive algorithm to compute the maximal information and corresponding optimal control: for $k = N - 1$ to 1, and $r = 1$ to R,

$$W^*(k, r, s_1, Z) = \max \left\{ \begin{array}{c} V^*(s_1, Z) + \mathbf{E}_s \left[q_1(s)W^*(k + 1, r - 1, s, Z_1) \right. \\ \left. + \, q_2(s)W^*(k + 1, r - 1, s, Z_2)\right], \\ \mathbf{E}_s \left[q_1(s)W^*(k + 1, r, s, Z_1) + q_2(s)W^*(k + 1, r, s, Z_2)\right] \end{array} \right\}. \qquad (1.15)$$

The logic behind (1.15) is the following. If a re-visit is utilized at the k^{th} object, we get the immediate payoff, $V^*(s_1, Z)$ plus the expected future return for $r-1$ revisits left at the next object, $k+1$. On the other hand, if the re-visit is not utilized, then we have r re-visits left at the next object and the corresponding expected return is the 2^{nd} term of the maximization in (1.15). We use the convention, $W^*(k, 0, s, Z) = 0$ for all k, s, Z and for the last object in the sequence, $W^*(N, r, s, Z) = V^*(s, Z)$, for all $r > 0$. Again, the optimal control,

$$u^*(k, r, s_1, Z) = 0, \text{ if } W^*(k, r, s_1, Z) = \mathbf{E}_s \left[q_1(s) W^*(k+1, r, s, Z_1) \right. \\ \left. + q_2(s) W^*(k+1, r, s, Z_2) \right], \tag{1.16}$$

and 1 otherwise. If $u^*(k, r, s_1, Z) = 1$, then the operator is presented with a 2^{nd} look of the k^{th} object at the optimal re-visit distance and angle, $d^*(s_1, Z)$ and $a^*(s_1, Z)$. Note that if the number of re-visits left is greater than the number of objects still to visit, i.e., $r > (N-k)$, re-visit will be utilized ("use or lose" scenario), and so, $u^*(k, r, s, Z) = 1, \forall s, Z$. It follows that all future objects will also be necessarily re-visited and there is no need to solve the recursion (1.15).

1.1.5 The Estimator.

The Estimator makes the binary decision of whether an object is a Target or a NonTarget by computing the probability of a Target based on the operator observations and operator error model using Bayes rule. This is shown below:

$$P(X = T | Y = T, s) = \frac{p P_{TR}(s)}{p P_{TR}(s) + (1-p)(1 - P_{FTR}(s))} \tag{1.17}$$

$$P(X = T | Y = NT, s) = \frac{p(1 - P_{TR}(s))}{p(1 - P_{TR}(s)) + (1-p) P_{FTR}(s)} \tag{1.18}$$

The posterior probability in (1.17) is stated as the probability that an object is a Target given that the operator said it was a Target at the state s. The posterior probability in (1.18) is similar, except the operator said it was a NonTarget. The lower case p is the prior probability or the Target density. The $P_{TR}(s)$ and $P_{FTR}(s)$ terms are the operator conditional probabilities, $P(Y = T|X = T, s)$ and $P(Y = T|X = T, s)$ respectively, contained in the operator error model. The model (table) entries are actually implemented as P_{MD} and P_{FA}.

The computed probabilities are compared to a threshold; if above the threshold, then the object is finally declared a Target. Example data for the decision threshold tradeoff is shown in Figure 1.1. The decision threshold is set at the a priori Target density P(T)=.5.

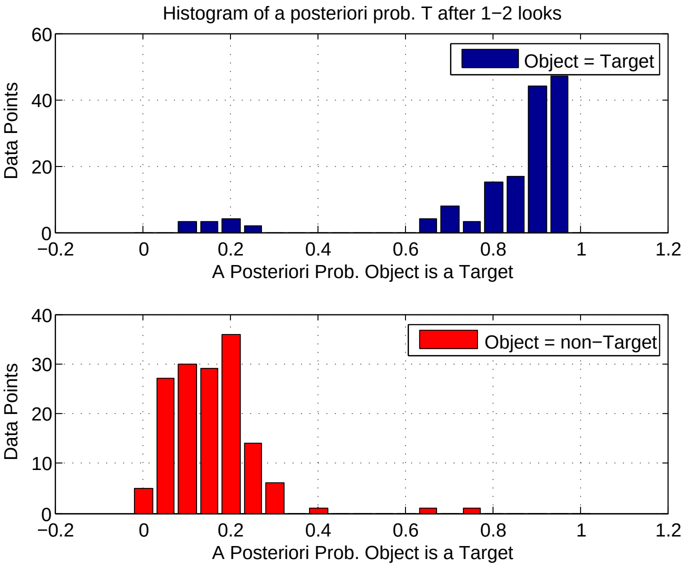


Figure 1.1: Errors as Function of Decision Threshold

The Targets to the left of .5 in the top plot are MD , and the NonTargets to the right of .5 in the bottom plot are FAs. As can be seen, there is no advantage in either FA or MD on moving the threshold out of the .4–.6 range.

In general, the threshold for this project is set at the Target density value. If there is only one observation, then this is the final decision. If there is a revisit, then there is an additional step to fuse the observations. This is done by taking the posterior in the 1st step and making it the prior for the 2nd step. The 2nd step, of course, uses the revisit state. The value is compared to the threshold, and the decision with 2 observations is complete.

Since the true class is known, the performance of the Estimator can be compared with the performance of the operator. Different performance metrics can be used, but percent correct (PC) is generally used in this report. Some alternative metrics are discussed in following sections.

1.1.6 Performance Metrics from Information Theory.

The inspected objects are $X = [T, FT]$, where the true classification is either Target (T) or False Target (FT). The un-assisted operator, or the SC–assisted operator, declares the inspected object to be a Target or False Target, $Y = [T_o, FT_o]$. The inspection operation is modeled as an asymmetric binary communication channel, as shown in Figure H.1.

The prior information, or density, is p, $P_{MD} = 1 - P_{TR}$, and $P_{FA} = 1 - P_{FTR}$.

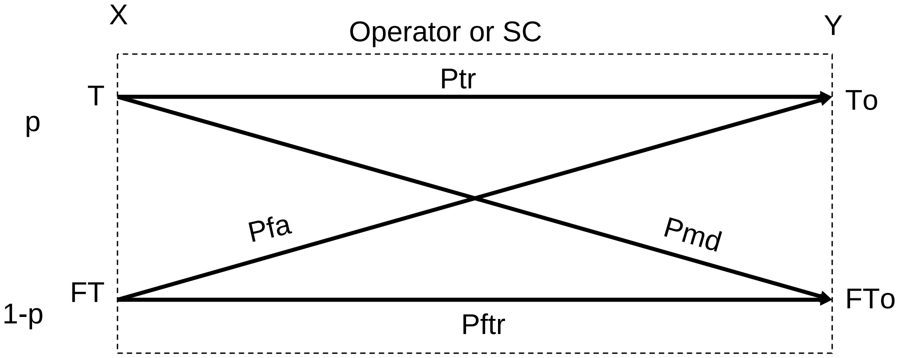


Figure 1.2: Binary Communications Channel

To evaluate the benefit afforded by the SC,

1. Evaluate the average mutual information according to equation [H.1], using the P_{TR} and P_{FTR} parameters of the operator, *and* the target density p (= 1/2). We obtain $I(X; Y)_o$.

2. Evaluate the average mutual information according to equation [H.1], using the P_{TR} and P_{FTR} parameters of the SC, *and* the target density p (= 1/2). We obtain $I(X; Y)_{SC}$.

3. Calculate

$$\Delta I(X;Y) \overset{\Delta}{=} I(X;Y)_{SC} - I(X;Y)_o \; [bits]$$

This is a replacement for the PC metric and also takes into account the target density p. Can also project ahead to account for different target densities, assuming that the operator's performance (P_{TR} and P_{FTR}) is not affected by the "knowledge" of p.

The Percent Correct (PC) is the probability of a correct transmission of a T or FT symbol being sent through the binary communication channel shown in Figure H.1. We calculate

$$PC = pP_{TR} + (1-p)P_{FTR}$$

By the same token, sending one symbol through the channel has a Probability of Error (PE) of

$$PE = pP_{MD} + (1-p)P_{FA}$$

Since $P_{MD} = 1 - P_{TR}$ and $P_{FA} = 1 - P_{FTR}$, then $PC + PE = 1$, as expected.

Remark If $p = 1/2$ then indeed $PC = (P_{TR} + P_{FTR})/2$.

In general, see Figure H.3

The PC (and PE) performance metric allows us to calculate the probability P of correctly classifying m objects when $n \geq m$ objects are being inspected. The number of correctly classified objects is a Bernoulli random variable and consequently

$$P = \binom{n}{m}(PC)^m(PE)^{n-m}$$

Note: If $0 < p \ll 1$ and, knowing this, the operator blindly adopts the strategy of classifying everything as a FT, that is, $P_{TR} = 0$, and $P_{FTR} = 1$, then $PC = 1 - p \approx 1$ and $PE = p \approx 0$. This creates an illusion of good performance. However, the information gain $I(X;Y) = 0$! This tells us that information gain might be a better performance metric.

As an example, take a $p = 0.05$ target density case where $PC = pP_{TR} + (1-p)P_{FTR}$: Op: FA=1%, MD=70%, PCavg=65%, whereas $PC = 0.05 * 0.3 + 0.95 * 0.99 = 95\%$

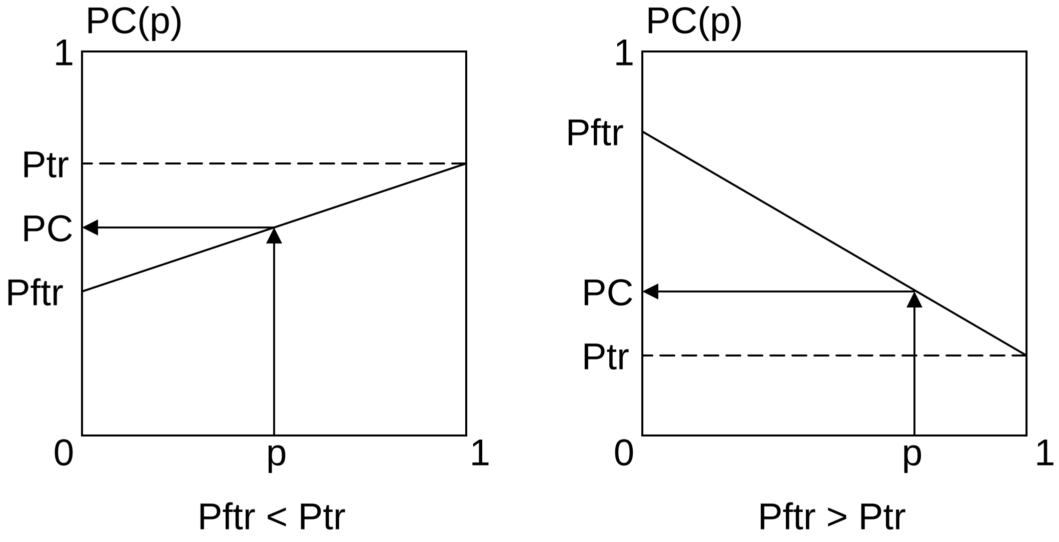


Figure 1.3: PC as a function of p

SC: FA=0%, MD=20%, PCavg=90%, whereas $PC = 0.05 * 0.8 + 0.95 * 1.0 = 99\%$

However, the $I(X;Y) \equiv$ Expected Information Gain performance metric:

$$\begin{aligned}
O_p : I(X;Y) &= 0.05 * 0.3 \log_2 \frac{0.3}{0.05 * 0.3 + 0.95 * 0.01} + 0.95 * 0.99 \log_2 \frac{0.99}{0.95 * 0.99 + 0.05 * 0.7} \\
&+ 0.05 * 0.7 \log_2 \frac{0.7}{0.95 * 0.99 + 0.05 * 0.7} + 0.95 * 0.01 \log_2 \frac{0.01}{0.05 * 0.3 + 0.95 * 0.01} \\
&= 0.0451928 \; [bits] \\
SC : I(X;Y) &= 0.05 * 0.8 \log_2 \frac{0.8}{0.05 * 0.8 + 0.95 * 0.0} + 0.95 * 1.0 \log_2 \frac{1.0}{0.95 * 1.0 + 0.05 * 0.2} \\
&+ 0.05 * 0.2 \log_2 \frac{0.2}{0.95 * 1.0 + 0.05 * 0.2} + 0.95 * 0.0 \log_2 \frac{0.0}{0.05 * 0.8 + 0.95 * 0.0} \\
&= 0.206196 \; [bits]
\end{aligned}$$

This needs to be compared to the channel's capacity, C_{O_p} and C_{SC}.

Note: When the operator adopts the strategy of declaring all inspected objects FT ($P_{FTR} = 1$ and $P_{TR} = 0$), then $I(X;Y) = 0$. If the operator uses a p loaded coin, then $I(X;Y) = 0$ also.

In general, given the operator or SC data P_{TR} and P_{FTR}, can calculate p^* and thus can calculate the channel's capacity C; we here contend that the operator or SC act as a communication channel. We have modeled the operator and/or SC as a binary communication channel and a natural performance metric is its capacity

$$C = \max_{p} I(X;Y)\ [bits]$$

Thus, $C = C(P_{TR}, P_{FTR})$ and is no longer dependent on the prior information/target density parameter p. Hence, calculate $C(P_{TR_{Op}}, P_{FTR_{Op}})$ and $C(P_{TR_{SC}}, P_{FTR_{SC}})$ and show $C(P_{TR_{SC}}, P_{FTR_{SC}}) > C(P_{TR_{Op}}, P_{FTR_{Op}})$.

Additional details are contained in the Appendix H "Communication & Information Theory".

This report describes a series of experiments related to the stochastic controller. (50% density, 5% density, Signal Detection Theory (SDT) modeling, Robustness test)

1.2 Background - ATR-Human Team

For the research described in this section [8], the objects of interest are classified by a UAV's Automatic Target Recognition (ATR) module according to the recognition of the presence of a certain feature. We assume that targets carry this feature, which is only visible (and hence, recognized) from the side and from a rather narrow range of AAs. When the pose of the object is such that the feature is visible, the ATR of the UAV will correctly report the object as a target. Unfortunately, the viewing angle of the UAV may not be suitable for recognizing the characteristic feature of the target in the image. In such cases, the ATR module does not declare the object as a target, and instead, the image of the object is transmitted to the operator for further examination.

The premise of this effort is that a human can classify objects significantly better than machines. One can then envision an operator-aided ATR scheme to enhance the performance of mission critical classification tasks. Even though the viewing angle of the UAV may not be ideal when recording an image of the object, a human is believed to use subtle cues for discerning certain features of interest. In such a case, the classification will be significantly enhanced by including an operator in the classification loop. This operator-aided ATR scheme is referred to as a mixed initiative system. It is assumed that the operator introduces a random delay in reporting his observations to the UAVs, and that his performance (in terms of delay and accuracy of classification) deteriorates when his workload increases—for example, when more than say four images per minute are received from the *m* UAVs.

The fuel on-board every UAV is a resource that must be optimally used. Each UAV carries a fuel reserve *r*, which is the fuel in excess of what is required to fly over the objects of interest assigned to it. In those instances where, due to the object's pose angle, the UAV ATR module may not be in a position to discern the feature carried by the target and consequently report it as a target, the UAV may replan its schedule upon receiving the report from the operator. Instead of continuing on to the next object of interest, it can turn around, revisit the object and obtain an additional view of the object from a better aspect angle. Obviously, the fuel cost associated with this action will be expended from the reserve. If the fuel cost is higher than the reserve, such an action cannot be taken as the UAV will not be able to visit all the objects assigned to it. The decision problem that naturally arises is the following: Given the communication delay (which dictates the cost of fuel associated with revisiting the object of interest) and the report of the operator about the object, should the UAV go to the next object in the sequence or should it turn around and revisit the object? The answer clearly depends on the information to be gained by revisiting the object, as well as the objective function for the optimal use of the resources.

The value of the information gathered from a 2nd view depends on how close the observations about the object reflect the status of the object. An assignment of an expected value to such an information at the time of decision making requires that the entire inspection chain be modeled. Given a model and the algorithms for optimal decision making, one can quantify the potential benefits to be had from cooperative target classification performed by the UAV/operator team. Some of the data required for this task includes: target density p; ATR confusion matrix; range of aspect angles to view feature; operator confusion matrix; and the statistics of the operator's delay τ.

The solution approach uses stochastic dynamic programming. This takes into account the probabilities of the ATR module and the operator misclassifying objects (his confusion matrix). The data required by the model is derivable from physical considerations and controlled human effectiveness experiments. For this reason, a realistic stochastic model of the surveillance mission and the associated decision process may be considered reasonable. A sequential inspection problem is formulated with elementary probability theory and a closed form solution of a stochastic dynamic program is used to obtain an optimal inspection strategy. An analytical approach is developed, with a view to capture and identify the critical target environment, the UAV error matrix, and the operator classification performance parameters that impact the effectiveness of the mixed initiative operations.

1.3 Background - LEAP II

The Learning Estimates of Aggregate Performance (LEAP) II was a joint effort of AFRL/RQQA with Aptima, Inc [11]. Prior to this effort, the error models were quite simple, being either a constant or a simple function of a single variable (see previous section). Here, the objective was to develop an operator error model which was a non-trivial function of at least 4 variable states. This necessitates the use of an operator-in-the-loop simulation to obtain the requisite data to be able to construct the error model. The Vigilant Spirit Control Station (VSCS) simulation environment is discussed in the next chapter and will not be belabored here. Various analyses were performed to determine the 4 variables to which the operator target classification error performance was the most sensitive. This turned out to be GSD, AA, DT, and IET. An important point is that these are dynamic variables in the simulation and are representative of random variation. This was discussed above. The Stochastic Controller (SC) then, is also a function of these 4 variables. The operator error model is also a strong function of the target density, which is discussed in later chapters. However, the target density does not vary during the simulations (fixed at 50%), and therefore the controller is not a dynamic function of target density.

There are a number of issues involved with operator-in-the-loop testing. One issue alluded to is the target density. A 50% density is not likely to be encountered, while 5% or 1% would be more reasonable. However, taking operator data at 5% density means that the number of target observations is only 1/20 of the number of nontarget observations. This means that the operator would have to perform 20 times the number of nontarget tests than would be needed for a 50% density experiment. It is impractical to perform that number of operator tests. Even at 50% density, there were 3,200 test samples that needed to be performed. This came from the decision to limit the test conditions to high/low values. This gives $2^4 = 16$ discrete "states" at which data is to be taken. A 100 samples per state objective was decided upon that should give a relative frequency accuracy of a few percent for False Alarm (FA) rate and Missed Detection (MD) rate at each state.

The decision was made to build an aggregate operator error model from a pool of test subjects. For one thing, the 3200 test cases would take many hours, which may be impractical for a single subject. For another, a single subject may be able to learn the parameters of the simulation and therefore bias the results. Finally, if a single subject is used, then the model is essentially tuned to the biases of that person and the model would not be representative of the expected average performance. For these reasons, 26 subjects were selected over a broad age and experience range. This gave 128 samples per operator times 26 operators to give 3328 total samples. After pruning the data, this resulted in 3179 samples for the functional model. The aggregate data for the operator error model gave an overall FA=32%, MD=13%, and PC=77%. Interestingly, these aggregate results are quite similar to the overall results from Operator S1 discussed in the section Test 1 below.

The data was used to train a statistical model using Gaussian Processes (GPs). GPs are a supervised machine learning technique that produces a posterior distribution over functions that best describes the data. This yields 2 models, one for FA and the other for MD. Both are a function of the 4 independent variables/states. These models closely approximate the data and provide estimates for intermediate states where there was no data taken. This model was then used with the SC in a validation experiment with 11 different test operators. The validation experiment was performed with a 5% target density, whereas the error model was trained with 50% target density. The result was that the majority of the validation operators had a FA rate in the single digits. This resulted in significant modeling error for those operators. At the time, the extensive conservative bias of the majority of the operators at low density was not fully appreciated. Only later, in the Integrated Human Behavior Modeling and Stochastic Control (IHBMSC) project for the Test 3 at 5% target density (see following Chapters) did the operator bias to extremely low FA rates become clear.

Also, it is interesting that the LEAP II operator error model was sufficiently similar to Operator S1 (see chapter on Test 1 below) that it could be used to initialize the SC for the data collection. This even though the scenarios and experiment had significant changes. The robustness was not expected, nor was the robustness of the SC to density changes expected (see Test 2 and Test 3 below).

1.4 Background - Signal Detection Theory

Signal detection theory has been used to model human behavior in many visual tasks including: monitoring a system with automated alerts [10], visual search/attention [13], and target acquisition [1]. Signal detection theory is also applicable to the *Target/Non-Target* decision that operators were asked to make in this study, which is equivalent to a yes/no decision about presence of a signal. The advantage of signal detection theory in this work is its ability to characterize two aspects of a decision: sensitivity and bias.

Sensitivity refers to the perceptual and cognitive capabilities of a human to distinguish between different states of the world. Within this study, there are two possible states of the world: a *target* letter is present or a *non-target* letter is present (also called target absent). In some circumstances the target and non-target letters can be easily confused, such as when the letters are very small in the video because the UAV's altitude is very high. Circumstances of high confusability are instances of low sensitivity. Other circumstances in which the letters are rarely or never confused are instances of high sensitivity.

Sensitivity is distinguished from bias, which is an overall tendency to respond *target* or *non-target*. Bias refers to the cognitive influences of the human's decision process. Bias is commonly described as liberal (saying *target* more than *non-target*) or conservative (saying *non-target* more than *target*). Bias may shift depending on: the likelihood of targets and non-targets, the goal of the operator (e.g., never miss more than 1/20 targets), or the costs and payoffs for decision outcomes. Following with early signal detection theory, we assumed that bias is independent from sensitivity. We also assumed that the experiences of a particular event were normally distributed and that the distributions of experiences had the same variance. (The question of measuring unequal variances or non-normal distributions requires high precision and a very large number of trials [4], making it more of a theoretical concern than a practical one).

In these tests there were four types of decision outcomes, resulting from combinations of the two states of the world and the two possible responses (Table 1.1). Signal detection metrics combine the information from hits (i.e., trials in which the target was present) and false alarms (i.e., trials in which the target was absent) to calculate sensitivity and bias. This combination of information from target and non-target trials is a distinction from the confusion matrices used in the SC's operator error model. A common metric of sensitivity is *d*', which can be visualized as the distance between the target and non-target distributions in z-score units (Figure 1.4). This metric can also be expressed in terms of maximum percent correct, which is the best performance that can be obtained for a given sensitivity. (Maximum percent correct is different from actual percent correct because maximum percent correct assumes an optimal bias, though in practice the two are often within 4-6%; [4]). A common metric of bias is β, which is directly related to the likelihood ratio between the target and non-target distributions.

Table 1.1: Decision Outcomes in Signal Detection Theory

	State of the World	
Response	Target	Non-Target
"Target"	Hit	False Alarm
"Non-Target"	Missed Detection	Correct Rejection

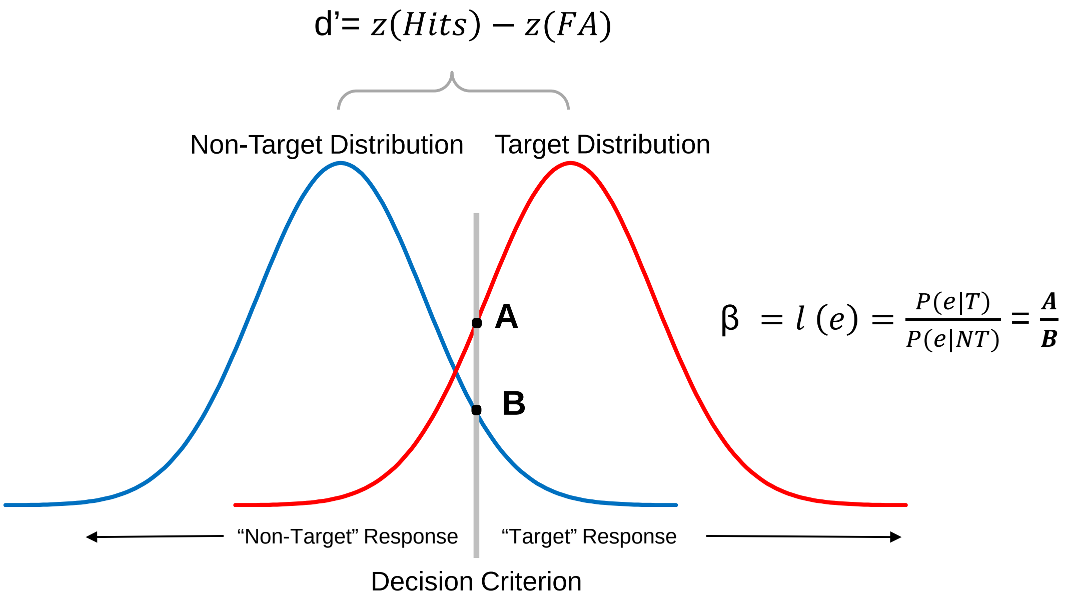


Figure 1.4: Theoretical Distributions of Target and Non-Target Events

II. General Method

2.1 Scenario

2.1.1 Operator Task.

The Reconnaissance, Surveillance and Target Acquisition (RSTA) sequential inspection task required the operator to view an object in a video and classify the object as a target or a non-target. Two seconds before the object entered the field of view, an alert appeared on the interface to signal the operator. The operator then responded either Target or Non-Target by pressing a button on the interface. If the operator did not respond, the cue extinguished 10 seconds after the object left the field of view. Operator confidence ratings were collected after each Target/Non-Target response. Test subjects spoke their confidence rating to the test administrator, who recorded it. Confidence ratings were collected for a number of reasons: 1) to investigate the relationship between operator confidence and performance, 2) to enable some bias analyzes with signal detection theory that were not possible with only Target/Non-Target responses, and 3) to assess their potential to further improve system performance if they could be incorporated into the Stochastic Controller (SC). For example, if the SC was not revisiting Points of Interest (POIs) that the operator had low confidence in their decision and their confidence was related to his/her accuracy, incorporating confidence ratings into the SC logic could better allocate the revisit budget.

In the task, the subject was instructed to "do the best you can" (i.e., maximize accuracy) and no performance feedback was provided during training or testing. The subject was instructed to sit with her eyes 32" from the monitor. No chinrest was used, but the distance was indicated with a marker and monitored by the test administrator. Other instructions given to the subject are described in Appendix D. There were 20 objects located at 20 POIs per run. Half of the objects were Targets and half were Non-Targets. In addition, 12 POIs were revisited in each run because the SC was incorporated into the system during all testing. The initial views of POIs combined with the revisits resulted in 32 observations per run. Each run lasted approximately 13 minutes. Between runs, the subject was given a 3-minute break that also allowed the simulation software to be reset with a different scenario. Test sessions never entailed more than 5 runs at a time, and typically one test session was conducted per day. On the few days that had two test sessions, one was held in the morning and one in the afternoon to allow a rest period of more than an hour.

2.1.2 The Target/NonTarget Models.

The Target is shown in Figure 2.1 and the NonTarget is shown in Figure 2.2.

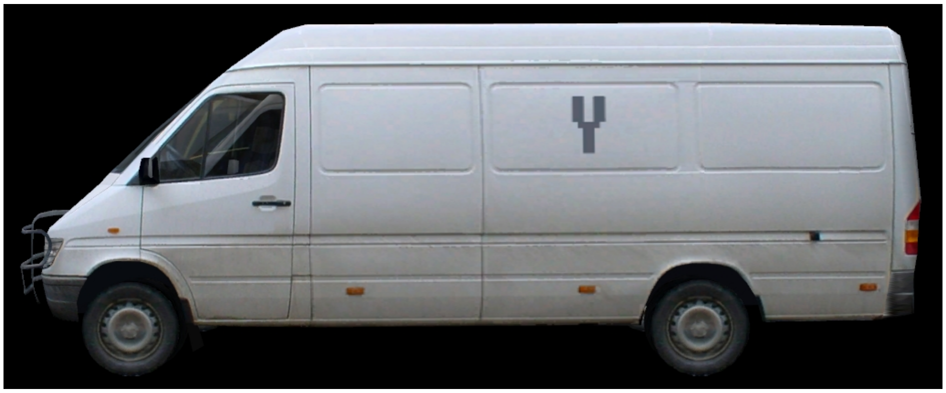


Figure 2.1: Vehicle with Target Feature Y

The vehicles with their respective letters established a classification task where the operator had to discriminate between a Target and a NonTarget. The scenario design objective was to find 2 features that were sufficiently similar that the operator had appreciable errors (FA,MD) on average. After pilot testing, the **Y** (Target) and a **V** (Non-Target) were settled upon (see Appendix A). Through additional pilot testing (See Appendix E) the chosen stimuli proved to be a moderately difficult classification task,

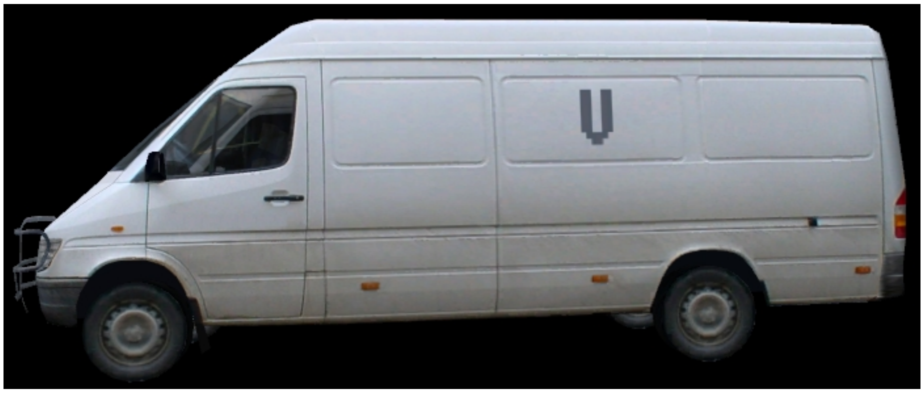


Figure 2.2: Vehicle with NonTarget Feature V

where the operator would have both misses and false alarm errors (Type I and Type II errors). The features of the letters were symmetrical about the vertical axis, and therefore the discriminability was not a function of direction, just the aspect angle (see Figure 2.6). The features were not symmetrical about the horizontal axis, but the nominal look angle of the sensor (depression angle) was fixed. This symmetry meant that approach direction did not have to be uniformly distributed over the sample states used for the operator model.

2.1.3 Stochastic Process Variables.

The scenario had four variables that defined the condition under which the Unmanned Aerial Vehicle (UAV) flew by an object (i.e., "state"). These variables were: Ground Sample Distance (GSD), Aspect Angle (AA), Dwell Time (DT), and Inter-Event Time (IET).

Ground sample distance is a measurement of the resolution of a digital image referring to how much distance of ground is covered in one pixel. A smaller ground sample distance corresponds to a higher resolution image. Ground sample distance was manipulated by fixing the camera zoom of the UAV and changing the altitude of its flight path. Ground sample distance, in combination with the constants of display size and the operator's distance from the display (32”), effectively manipulated the visual angle of the letters (i.e., their size on the retina).

Aspect angle is the angle in two-dimensional space between the vertical side of the van displaying the task letters and the UAV’s flight path.

Dwell time is a measurement of how long the task letter was presented within the simulated UAV sensor (i.e., the observation interval). Dwell time arose from the time it took the letter to move across the sensor’s field-of-view, which was a fixed distance on the display monitor. Therefore, dwell time co-varied with the letter’s velocity on the display monitor.

Inter-event time is a measurement of how much time elapsed from the end of one observation to the start of the next observation (i.e., inter-stimulus interval). Inter-event time was used as a way of effecting operator workload, analogous to the event rate in vigilance tasks (for a review of event rate and workload in vigilance, see [14]). Inter-event time was such that it could be negative (in other words, observation intervals could overlap). In these instances, inter-event time theoretically has some relationship to dwell time in that when a letter is present in the videos of two different UAVs at the same time, any viewing strategy will reduce the actual length of observation of one or the other (or both) letters.

Table 2.1: Ranges of Independent Variables

	GSD	AA	DT	IET
Minimum	0.1909 $\frac{ft}{pixel}$	45°	2.24s	0s
Maximum	0.2757 $\frac{ft}{pixel}$	90°	4.55s	500s

A planner was used to generate a UAV trajectory to visit all the objects. The stochastic aspects of the UAV flight path that affected GSD, AA, DT, and IET were determined by analyzes in the LEAP II program. These variables were randomly sampled from uniform distributions to simulate the stochastic nature of realized flight path (Table 2.1). For revisits, the premise was that once the UAV flew over the object and the object was in the image frame, then the true state could be determined. For example, if the size of the object was known a priori, then the UAV height above terrain (Above Ground Level (AGL)) could be computed as well as an altitude bias *at that location*. Also, knowing the ground speed, airspeed, and heading, the local wind vector could be estimated. Given the local altitude bias, the actual object pose angle, and the local wind vector, then a revisit state command could be computed that yielded the realized optimal revisit state computed by the stochastic controller.

The simulation approximated the previously described procedure and uncertainty. For the initial object visit state, or "first look," the guidance algorithm or planner (Cooperative Control Algorithms (CCA)) sampled the altitude, aspect angle, and speed (dwell) uniform distributions to yield the commanded and realized state. For the revisit, or "second look," the optimum state commanded from the stochastic controller was the realized state. The simulation closely achieved the commanded states.

2.2 Apparatus

The apparatus was the Vigilant Spirit Control Station (VSCS), developed by 711HPW/RHCI. It is a research test-bed used to develop, integrate and test advanced technology for single-operator, multi-UAV RSTA tasks and missions. VSCS served as a single interface for performing both piloting and sensor operations and operators interacted with it through a mouse and keyboard. For this set of studies, we used a customized version of the VSCS that consisted of two windows (Figure 2.3). Each window was displayed on its own 24" monitor at a screen resolution of 1920 x 1200 (Dell 2408WFPb; Round Rock, TX). On the left window was the Tactical Situation Display (TSD) which showed the real time position and flight plan of the UAVs overlaid on a map. On the right window were the individual video feeds from each UAV.

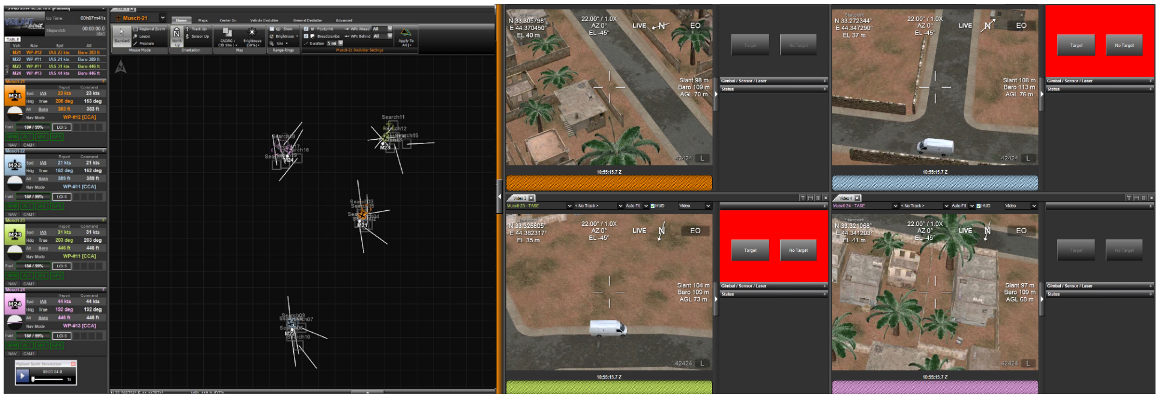


Figure 2.3: Tactical Situation Display and 4 UAV Video Display

Tactical Situation Display

The TSD is shown on the left side of Figure 2.3. The operator could see the 4 UAVs and their point search tasks at a compressed scale. No useful detail could be seen at this scale.

In general, the operator could have chosen whatever scale he or she wanted, with however much detail desired. At the exploded scale, the operator would have been able to predict when the next UAV was going to fly over an object. However, this was rendered unnecessary by the alert (cue). Ultimately, the exploded view was not used by the operator.

Video Display

The video display is shown on the right side of Figure 2.3. It shows the simulated feeds streamed from each of the UAVs. The video images were generated using MetaVR©. There was a 3-dimensional terrain database and cultural objects defined for the images. The individual UAV dynamics and sensor motion were simulated on the VSCS. UAV positions and states were continuously streamed to the graphic computers to define a viewing state into the 3-dimensional database. The view at this state was then sent to the VSCS and displayed on the individual video panes. The Target/Non-Target buttons were located to the right of the corresponding video. The alert appeared behind the buttons as red fill and is illustrated in two of the video panes. A number of issues occurred during video development (see Appendix B) but were resolved to satisfaction before the tests occurred.

The Terrain Data Base

The terrain database had Level 2 digital terrain elevation data for the Demo Baghdad area. There were high resolution textures and cultural objects in a small urban section. The detail area was replicated (i.e., tiled) so that the UAVs were spread out over a large area and didn't overlap. Target/Non-Target models were defined and placed uniformly, with some randomness, in the detailed sections of the environment. The vans were stationary for the simulation trials.

2.3 Cooperative Control Algorithms

The Cooperative Control Algorithms, or CCA, had several functions, but the most relevant to the Integrated Human Behavior Modeling and Stochastic Control (IHBMSC) project was the path planning and the SC. A top-level diagram of the Cooperative Control Algorithms (CCA) is shown in Figure 2.4 below. The class of task used here was the Point Search Task (PST). These tasks constituted part of the mission objectives.

The point search tasks could be entered by the operator or from a file. A point search task consisted of the Longitude, Latitude, Mean Sea Level (MSL) altitude, and the Heading at which the point was to be viewed. Also needed was the Unmanned Aerial Vehicle (UAV) speed and UAV altitude, as well as the UAV sensor footprint model. See Appendix C for the detailed UAV specification. Each of the 4 scenarios had 20 point search tasks for the 20 vans. There were 4 UAVs available to perform the point search tasks, and the tasks were set up so that only 5 tasks were assigned to a UAV. It was important that only 5 tasks be assigned, because the Stochastic Controller (SC) was set for a sequence of 5 objects. This was not intrinsic to the algorithm, but did greatly simplify data gathering and analysis.

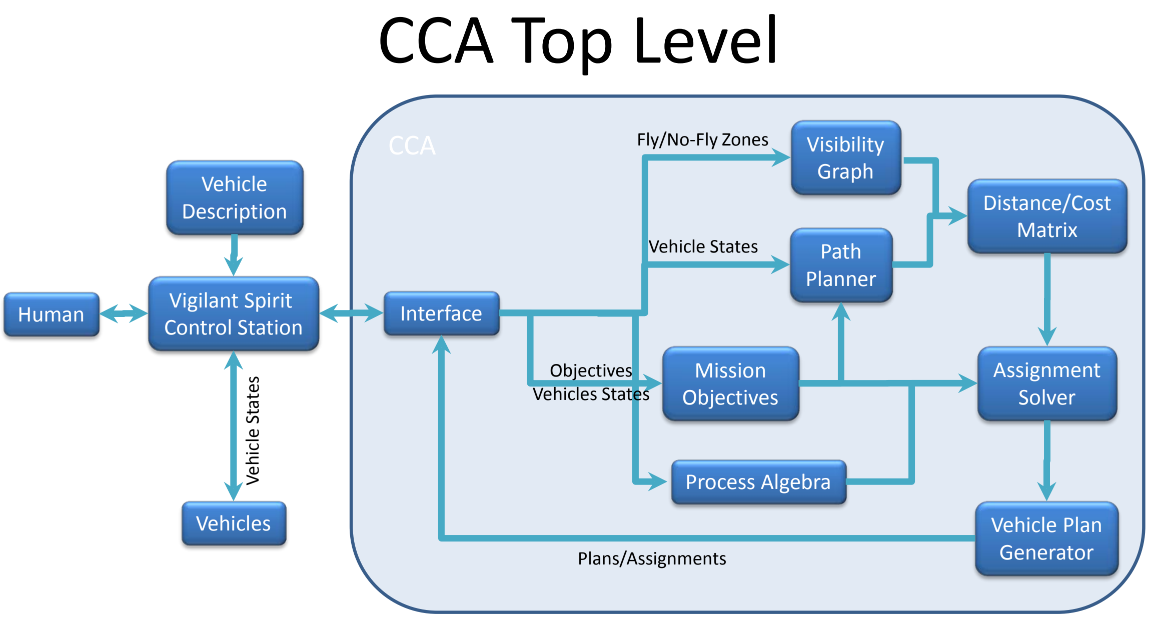


Figure 2.4: CCA Architecture Diagram

The CCA planned a path for each of the 4 UAVs to visit the allocated 5 objects. These 4 paths were completely independent—there was no coupling between the UAVs. This greatly simplified the design of the planner and the SC. The planner tried to find the minimum total Dubins path length between the 5 objects (vans). This was a sequence of straight lengths and arcs, where the arc radius was determined from the desired turn rate.

The plan could have had additional constraints, but the most important here was for the sensor. There was a stand-off, or a minimum distance from the object, that the UAV was on a straight (not arc) path. As the sensor was body fixed, this ensured that the sensor footprint was on the path when flying over the object. Also, there was a constraint that the footprint had to be completely past the object before the path could turn to the next object to visit.

Stochastic Controller

The stochastic controller architecture is shown in Figure 2.5. The SC module was part of the CCA. The SC consists of several modules, and included: the design of the control matrix; the interrogation of the control matrix; the revisit planner; the operator alert manager; and the output data at the end of the scenario. The control design algorithm took as input: the 5 inspection tasks per UAV; the operator model; the density; and the number of revisits. The true density was computed from a configuration file (`PointSearchTask.vsxml`) that identified the Points of Interest (POIs). The computation of density and CCA path planning made up the initialization of the SC. The control design

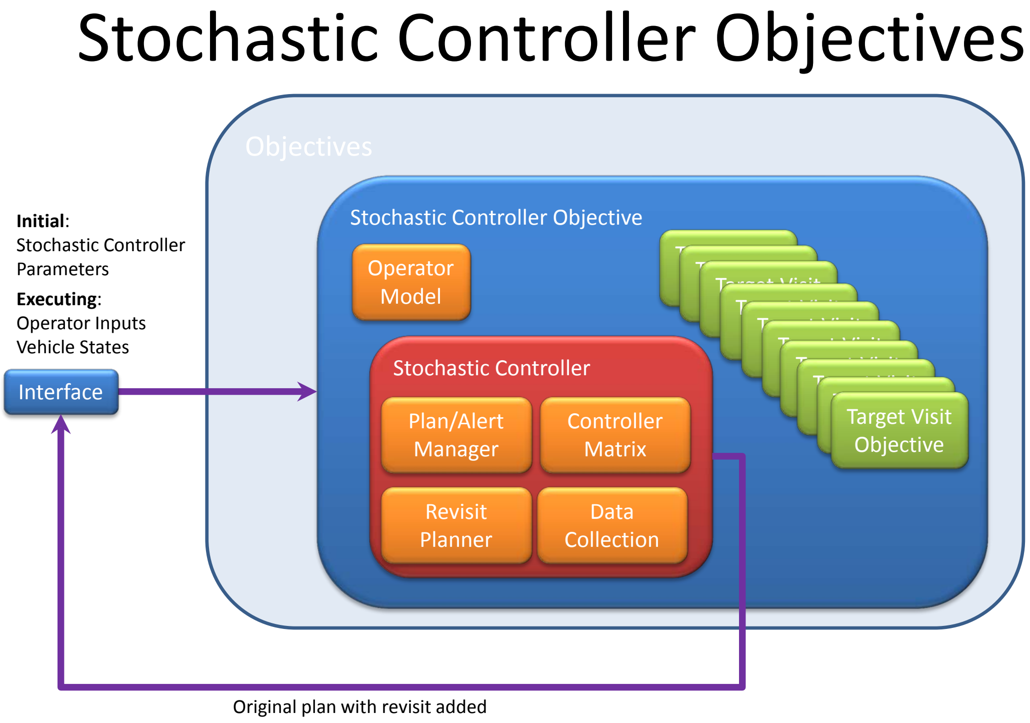


Figure 2.5: Stochastic Controller Architecture

algorithm was not strictly a function of the planner output. This was because the inspection length of 5 was fixed, the number of revisits was fixed at 3, and the weights were not a function of the distance traveled. Also, the control design was not a function of the initial visit state. For these reasons, the controller matrix was the same for each of the UAVs.

During the execution phase, the plan/alert manager monitored the progress of the UAV in cycling past the plan waypoints. During execution, the first thing to happen was a Type 3 event where the UAV passed the Alert-on waypoint. The manager sent a signal to the Vigilant Spirit Control Station (VSCS) to turn on the respective video alert. Next was a Type 1 event where the operator declared the object in the respective video was a Target/NonTarget. The operator input and the vehicle states were sent to the controller matrix, i.e., $u = f[GSD, AA, Dwell, IET, Operator, Revisit]$. The Ground Sample Distance (GSD) was computed from the known MSL of the object and the MSL of the UAV (MSLa-MSLb=AGL where GSD=f[AGL,sensor]). The Aspect Angle (AA) was computed from the UAV heading and the object pose angle (AA=Heading-Pose). The Dwell Time (DT) was computed from the difference in the footprint leading and trailing edge times passing the object (Dwell=f[AGL,speed,footprint]). The Inter-Event Time (IET) was computed as the elapsed time since an alert was set and was a function of all 4 vehicles (IET=f[AlertV1,AlertV2,AlertV3,AlertV4]). The target orientation is shown in Figure 2.6. As can be seen, the Pose Angle of the vehicle is the number of degrees off North. The Aspect Angle of the view is the number of degrees off the nose of the vehicle. For Figure 2.6, if the objective is to have the UAV fly over the object beam-on (AA=90deg), then the UAV heading is (PoseAngle=45deg) + (AA=90deg) + 180deg = 315deg.

Once the state [GSD,AA,Dwell,IET], the operator's response (Target/Non-Target), and the number of the remaining revisits were obtained, then the control matrix could be interrogated to yield the revisit decision. If $u = 0$, then the object would not be revisited and the UAV would proceed to the next object in the sequence. If $u \in \{1, \ldots, 256\}$, then the UAV was commanded to revisit the object at the respective state. This kicked off the revisit

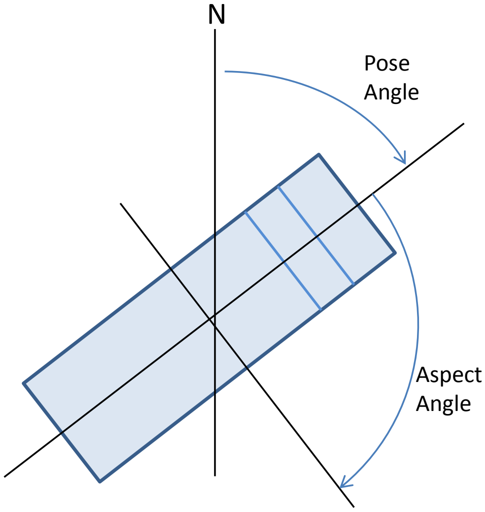


Figure 2.6: Target Orientation

planner, where a snapshot of a UAV revisit trajectory in the TSD is shown in Figure 2.7. Here, the white lines are the true beam-on headings to look at the 5 objects (van). The

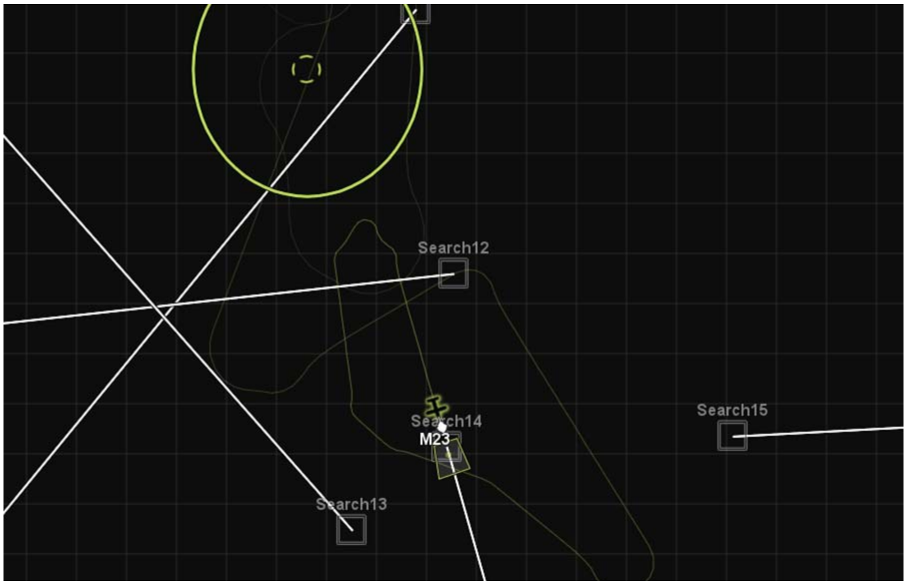


Figure 2.7: TSD view of UAV revisit

faint green line is the computed revisit trajectory. The small box in front of the UAV is the projected sensor footprint. When the revisit decision was made, a minimum length Dubins path was computed to view the object at the AA (and altitude) specified in the controller. Once the footprint has passed the object, a Dubins path was computed to intercept the original path to the next object in the sequence at the specified stand-off distance. This completed the sequence of events for inspecting an object.

At the completion of the inspection scenario, the data log file was written out. This file contained the 1st and 2nd look state, the operator input, plus the P_{FA} and P_{MD} at those states. This enabled the $P(T)$ to be calculated and the performance of the Estimator to be evaluated.

III. Tests at Target Density of 50%

3.1 Test 1: Single Operator Model Creation and Performance Assessment

This section describes an assessment of the Stochastic Controller (SC) as well as the creation of an operator model. For the assessment, we expected that the operator and SC combined (revisits and the Estimator) would perform better than the operator alone (performance without revisits). In order to conduct this assessment we needed to conduct human-in-the-loop test using an operator error model of that individual in the SC. We had to create that model prior to conducting our test since no such model existed. The operator error model in the SC had to capture how an operator's performance varies as a function of four variables of interest: Ground Sample Distance (GSD), Aspect Angle (AA), Dwell Time (DT), and Inter-Event Time (IET). We expected that operator error would be higher with higher values of GSD than with lower values and operator error would be higher with lower values of AA, DT, and IET than with lower values. No predictions were made concerning interactions between variables, however, we expected that model development could be made more efficient if no interactions were found between variables because future operator models could be developed by modeling the impact of each variable independently without needing to factorially combine variables.

3.1.1 Method.

3.1.1.1 Subject.

One member of the research team volunteered to participate as a test subject (S1). Operator S1 had normal vision and aged 21. She was experienced with psychological research tasks and was informed of the purpose of the study.

3.1.1.2 Task.

The subject was introduced to the interface and the task prior to beginning data collection. The subject was told that there were an equal number of targets and nontargets in each run. Previous research under the LEAP II project showed that the initial task was too easy for test subjects, leaving few errors for the SC to overcome. Extensive pilot testing was conducted using members of the research team (including S1) in order to find ranges of values for each independent variable that would avoid these past 'ceiling' effects. Examples of these pilot results for one team member are contained in Appendix E. Final ranges for variables are presented in Table 2.1. Prior to beginning data collection for the model, we checked that the subject's performance had not degraded below 70% (average percent correct) and that performance was stable and asymptoted (i.e., the test subject was done learning the task).

For the model creation effort, the subject performed 120 runs that each had 10 Target and 10 Non-Target trials (2400 trials overall). This number of trials would provide approximately 75 samples of each of the 16 modeled states (median-split combinations of all four variables) for target and non-target trials. Some technical issues with data logging software resulted in losing a small number of trials (<1%), leaving 2,395 trials for creation of the operator error model.

Important to note, model creation runs also included the SC in the system loop, meaning that they had 12 second look trials in addition to the 20 first looks. For the SC to function within the system loop during initial model creation runs, the operator model from LEAP II was used. This LEAP II model was replaced by an interim model for operator S1 after 25 runs and updated after 75 runs. These interim models are presented in Appendix F. The LEAP II model and the interim models were relatively similar and changing them did not appear to impact operator performance during testing. Once the complete 120-run operator model was created, an additional 10 runs were conducted using that model to assess the performance impact of the SC.

3.1.2 Results and Discussion.

3.1.2.1 Operator Modeling.

A total of 120 data runs (2,395 first look trials) were used to create the operator confusion matrix entries in *LEAPmodel0920.xml*. Overall performance on these trials was: S1: FA=35%, MD=17%, and PC=73.8%.

The final operator error model is shown in Table 3.1. The model shows the 16 different states that are based on median-split combinations of the 4 variables. Variable values of high and low are identified by their maximum and minimum, except for IET which had a maximum of 500 seconds. The stochastic selection of states resulted in an average of 74 samples per state with a Standard Deviation (std) of about 11 samples. Across states, FA rates ranged from 2.8% to 69.9% and MD rates ranged from 3.3% to 36.0%. A comparison between the final model and the interim model based on 75 runs showed that the individual FA and MD values varied less than 5%. This similarity implied that any addition of data to the model should have a small impact on the model.

Table 3.1: Operator Model Data F0920

State	GSD	Angle	Dwell	IET	Samples	Hits	FA	Samples	Hits	MD
1	0.1909	45	2.24	0	80	29	0.3625	57	47	0.1754
2	0.1909	45	2.24	30	74	30	0.4054	67	60	0.1045
3	0.1909	45	4.55	0	58	10	0.1724	71	61	0.1408
4	0.1909	45	4.55	30	66	13	0.1970	75	65	0.1333
5	0.1909	90	2.24	0	50	7	0.1400	63	53	0.1587
6	0.1909	90	2.24	30	64	10	0.1563	52	48	0.0769
7	0.1909	90	4.55	0	64	3	0.0469	68	62	0.0882
8	0.1909	90	4.55	30	72	2	0.0278	60	58	0.0333
9	0.2757	45	2.24	0	91	51	0.5604	89	57	0.3596
10	0.2757	45	2.24	30	83	58	0.6988	78	68	0.1282
11	0.2757	45	4.55	0	95	38	0.4000	92	67	0.2717
12	0.2757	45	4.55	30	72	35	0.4861	60	51	0.1500
13	0.2757	90	2.24	0	79	35	0.4430	99	78	0.2121
14	0.2757	90	2.24	30	78	40	0.5128	66	51	0.2273
15	0.2757	90	4.55	0	83	28	0.3373	109	86	0.2110
16	0.2757	90	4.55	30	81	25	0.3086	79	64	0.1899

3.1.2.2 Signal Detection Analyses of Operator Performance.

Further analyses of the operator's pattern of errors were conducted using signal detection theory. Analyses were conducted with "first look" trials only (n = 2,395), unless otherwise noted. The overall performance of the operator across the entire test was *d*' = 1.30, $\beta = 0.71$ (i.e., liberal). Sensitivity for high and low values of each variable is shown in Figure 3.1. Sensitivity is expressed as percent correct maximum and the variables have been median split (i.e., range of values of each variable has been separated into high/low categories about the median). Percent correct maximum represents sensitivity by assuming

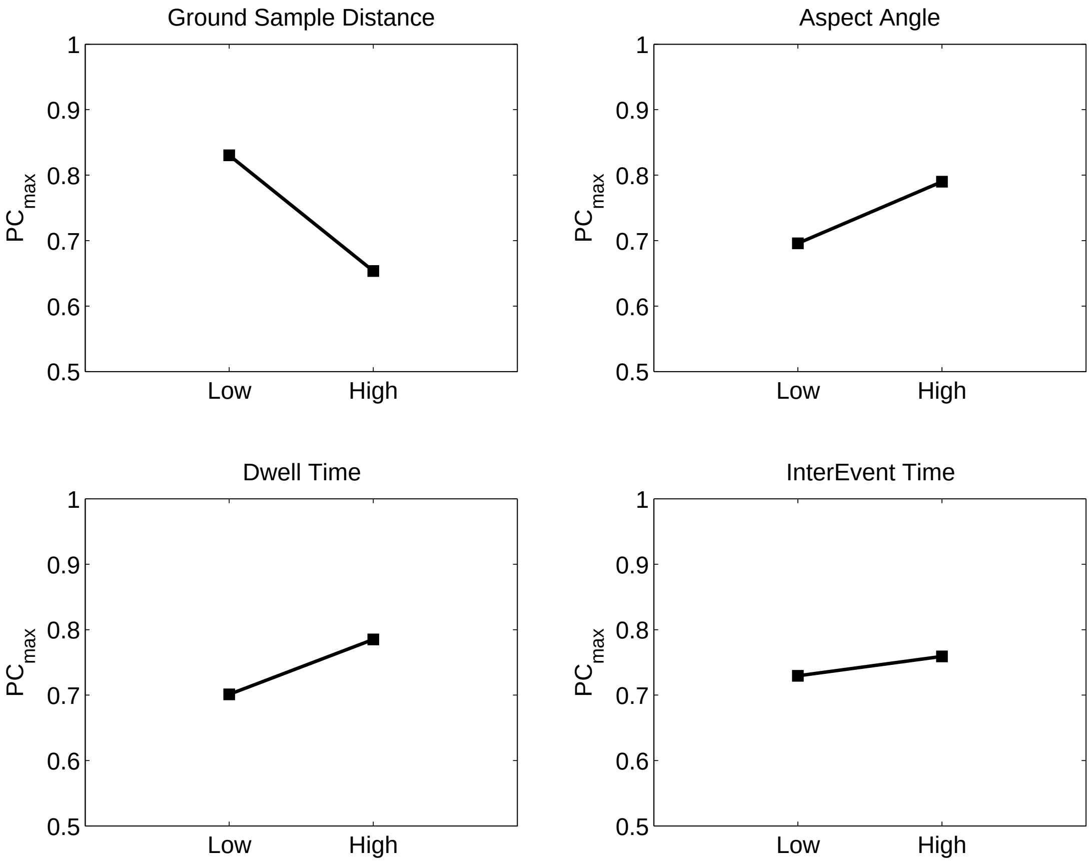


Figure 3.1: Operator Sensitivity for each Variable

an optimal bias that would maximize percent correct. For situations like this one, where targets and non-targets were equally likely to occur and there were equal payoffs and costs for all correct responses and errors, the optimal bias is 1. For the ranges sampled, all variables affected operators in the expected direction (as indicated by the direction of their slope). Low values of GSD increased performance compared to high values and high values of AA, DT, and IET increased performance compared to low values. Comparing between the slopes showed that ground sample distance had the largest effect on sensitivity ($\Delta d'$ = 1.12) and inter-event time had the smallest ($\Delta d'$ = 0.19).

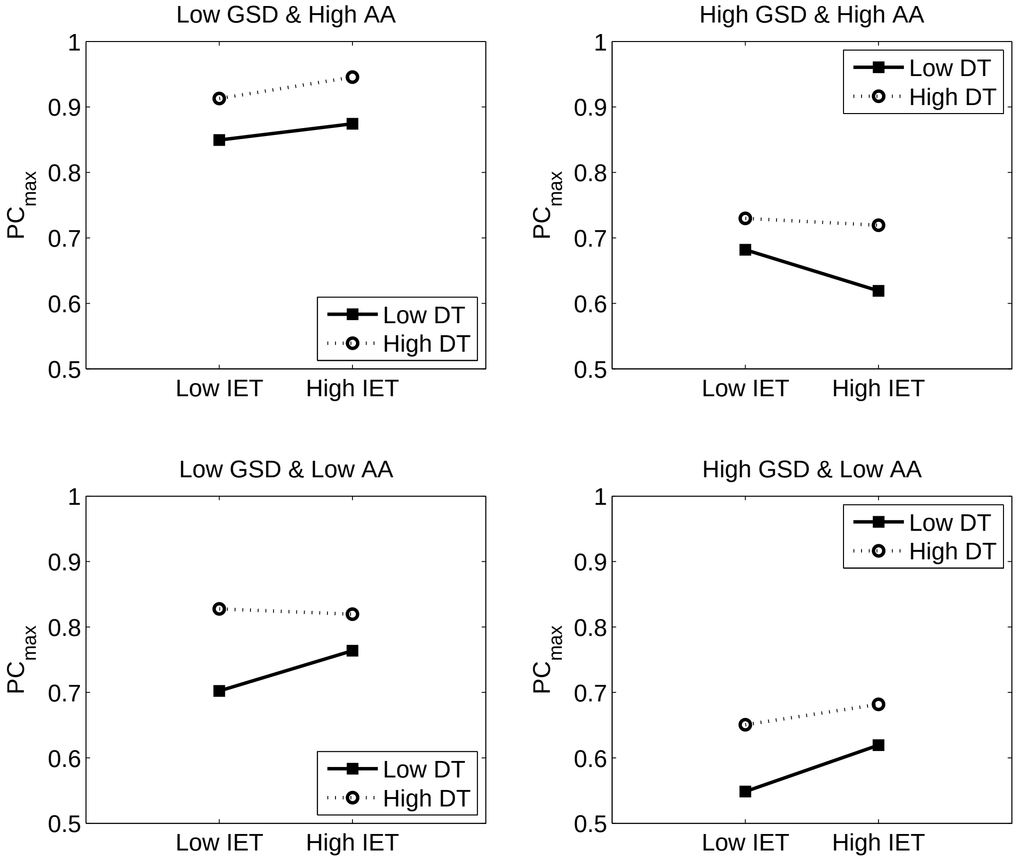


Figure 3.2: Operator Sensitivity for the 16 Model Bins

The combination of both levels of all variables is shown in Figure 3.2. Absence of interactions would be shown if both lines in each panel were parallel. The upper right panel and lower left panel show lines that are clearly not parallel, yet the direction of the interaction of DT and IET are in different directions. A combination of these effects would cancel out, possibly explaining the flat main effect of IET shown previously. The lines in the lower right panel are close to parallel. This panel also shows the worst state in terms of sensitivity, high GSD, low AA, low DT, and low IET. The lines in the upper left panel are parallel as well. This panel also shows the best state in terms of sensitivity, low GSD, high AA, high DT, and high IET.

To have better insight into the effects of the variables, we approximated psychometric functions using the first look data. Psychometric functions show percent correct performance (not to be confused with PC_{max}) across a range of variable values. Psychometric functions were approximated by repeatedly splitting each high/low group about its median (Figure 3.3). This figure shows orderly effects for GSD, AA, and DT, though the range of variable values was restricted. The effect of IET was less orderly than the other variables. A linear fit to IET yields a slope of almost 0 and an intercept at the average PC for all first look trials.

One IET that we were interested in specifically was the occurrence of overlapping observations–instances in which the operator would have to split their viewing time between multiple video feeds. We expected that overlap instances would impact performance because they are related to how much dwell time the test subject has to view the Point of Interest (POI). There were and small sample sizes of 3 or 4 simultaneous letters onscreen (41 trials, or 0.34%) because overlap instances and IETs in general were not designed into the stimulus conditions, as previously mentioned. To increase the sample size, all overlap trials, including instances of 2 simultaneous letters onscreen, were grouped

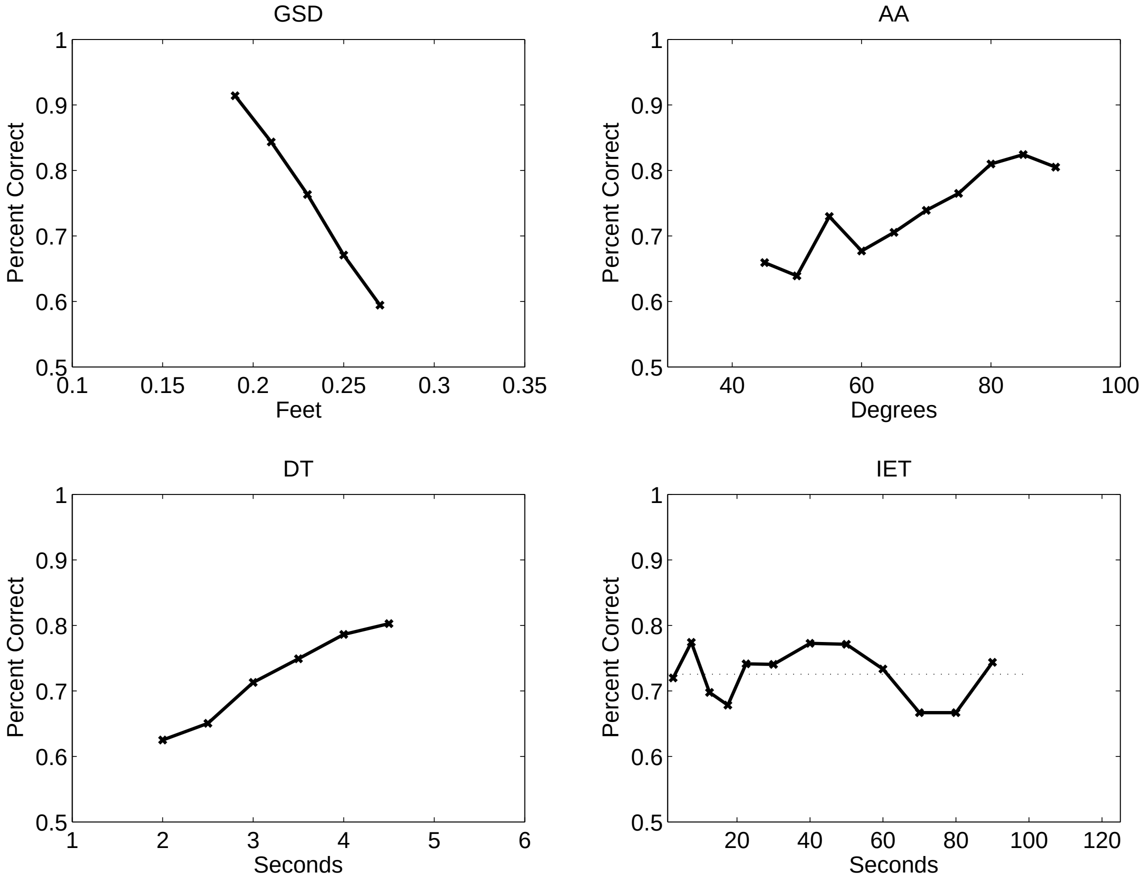


Figure 3.3: Approximated Psychometric Functions for each Variable

together (559 trials, or 4.66%). To further increase the sample size, we calculated a "generous" measurement of overlap, which began at the pre-observation cue that alerted operators to the upcoming inspection point and ended at the time of response (845 trials, or 7.05%). For all the ways we identified overlap trials, the difference in percent correct between overlap trials and all other trials was less than 1%. One difficulty in assessing the effect of overlap in this data set is that overlaps contained both first looks and second looks. Second looks were consistently very high in accuracy, and so reduced the variability in performance.

3.1.2.3 Analysis of Operator Confidence Ratings.

The confidence ratings are defined as: 1 = no confidence; 2 = low confidence; 3 = moderate confidence; 4 = high confidence; 5 = very high confidence. These ratings were provided by the operator for each Target/non-Target decision. A histogram of confidence ratings for first look trials shows that the most common confidence response was 2, followed by 1, which together accounted for 84.5% of responses (Figure 3.4). Confidence ratings of 3, 4, or 5 accounted for the remaining 15.5% of responses. During experiment design and pilot testing, the range of stimulus values was adjusted intentionally so that performance contained errors and the task was moderately difficult (in order to avoid floor and ceiling effects). The number of low confidence ratings confirm the stimuli selected were within the range of interest for this test.

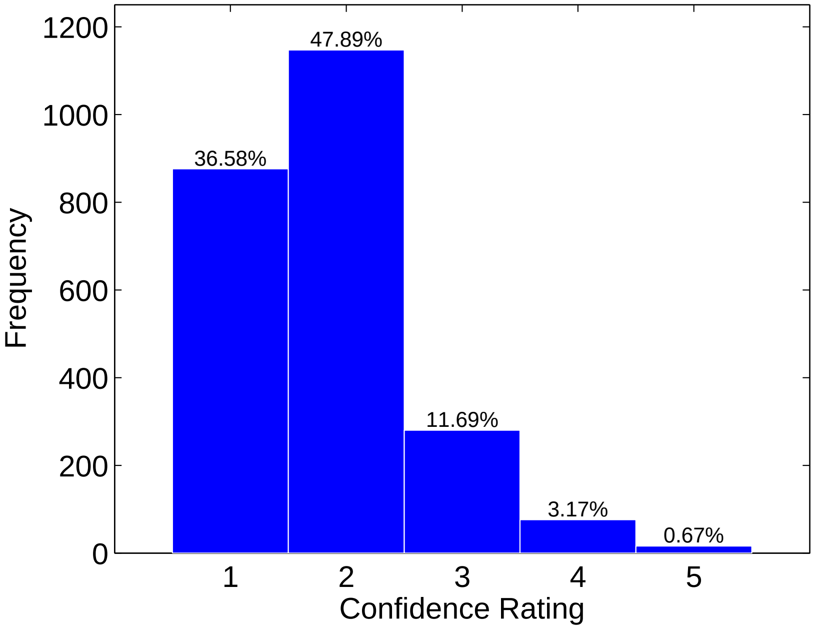


Figure 3.4: Histogram of Confidence Ratings

Figure 3.5 shows average performance as a function of the operator's confidence rating. Confidence was strongly related to performance. For confidence of 3, 4, and 5, performance was very high. Performance was low when confidence ratings were 1 or 2. Simple correlations between confidence responses and variable values showed that: GSD had a strong negative relationship to confidence (*r* = -0.61), AA and DT had medium positive relationships to confidence (*r* = 0.21 and 0.20, respectively), and IET had no relationship to confidence (*r* = 0.01). These correlations agreed with the sensitivity results presented in Figure 3.1.

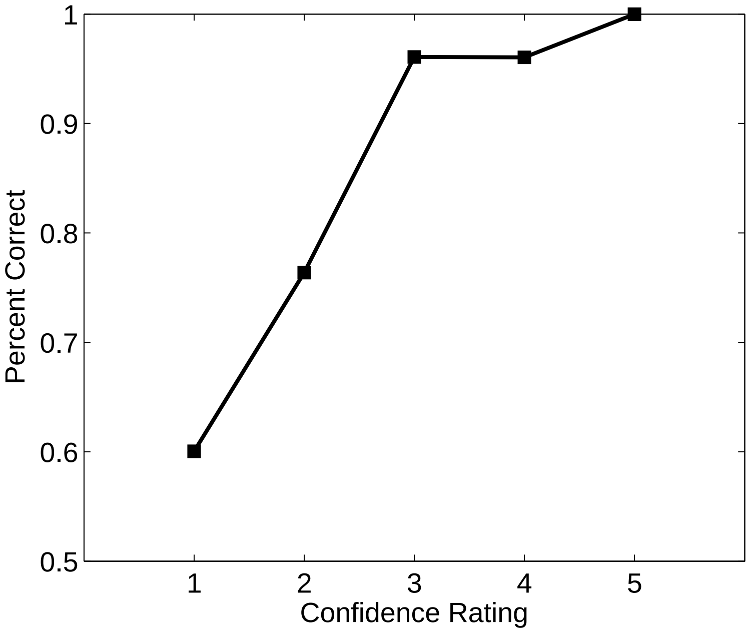


Figure 3.5: Relationship between Confidence Ratings and Performance

Confidence ratings were used to synthesize a "Type-II" Receiver Operating Characteristic (ROC) curve (Figure 3.6). An ROC curve plots hit rate against false alarm rate and typically shows performance across a variety of biases or decision criteria. Perfect performance would fall in the upper left corner and chance performance falls along the positive diagonal. Rather than measure different biases as in a Type-I ROC, Type-II ROC curves estimate different biases of the operator by recoding the operator's responses. The operator's confidence ratings for non-target responses were inverted so that all target and non-target confidence ratings were on a single scale ranging from very high confidence non-target (-5) to very high confidence target (+5). A liberal bias was synthesized by giving "target" responses to confidence greater than or equal -1 and a very liberal bias was synthesized by giving "target" responses to confidence greater than or equal to -2. A conservative bias was

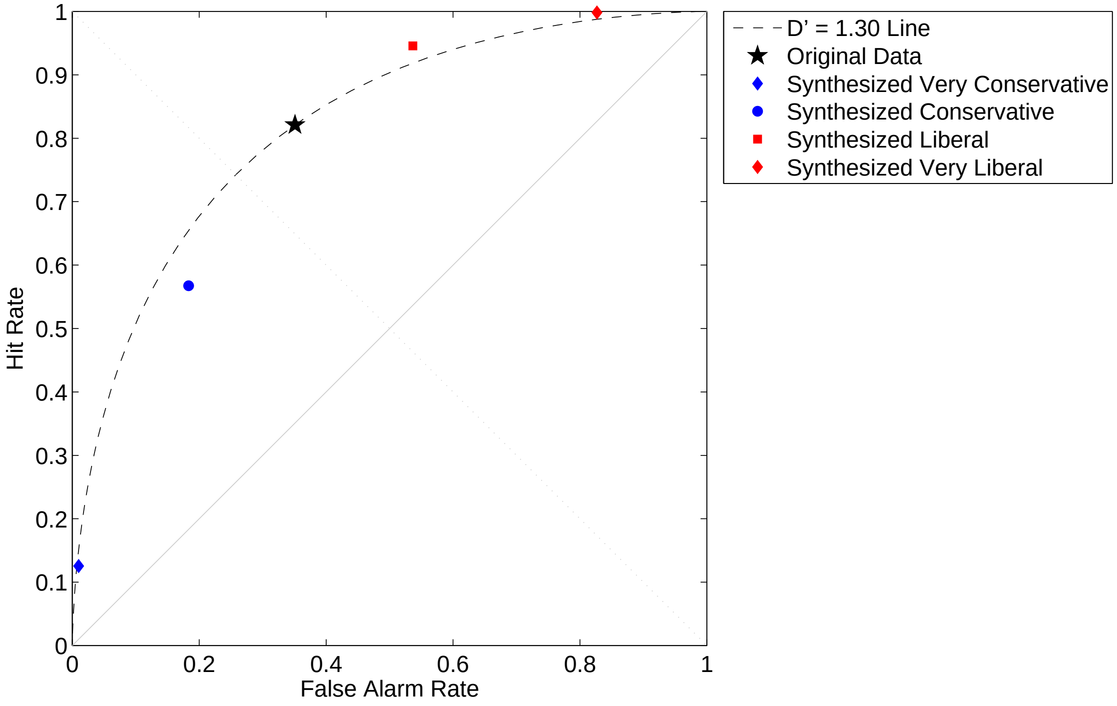


Figure 3.6: Type-II ROC Curve. See text for explanation.

synthesized by giving “non-target” responses to confidence less than or equal to +1 and a very conservative bias was synthesized by giving “non-target” responses to confidence less than or equal to +2. Confidence ratings of ±3, ±4, or ±5 were not recoded (due to their high accuracy). The plotted line is based on the overall observed *d*’ (1.30) and the assumption of equal-variance normal distributions, which yielded a good fit to the synthesized data. From this data, it appears that the signal detection model could be used for approximating the human’s performance across a range of different biases for future development of the SC. Being able to approximate the human’s performance would be useful particularly for instances when targets are extremely rare or extremely likely (e.g., 1% of Points of Interest (POIs) or 99% of POIs) because achieving large sample sizes through human testing is time consuming.

3.1.2.4 Stochastic Controller Assessment.

The results from the 10 assessment runs that used the final operator error model (*LEAPmodel0920.xml*) are shown in Table 3.2. The SC reduced both the False Alarm (FA) and the Missed Detection (MD) rates, resulting in a large increase in percent correct. The SC reduced the FA rate by nearly a factor of 8, reduced the MD rate by a third, and increased the Percent Correct (PC) by nearly 20%. The SC had a smaller impact on the MD rate compared to the FA rate, which may be due to there being fewer MDs to begin with.

Table 3.2: Results of Stochastic Controller Assessment

	FA	MD	PC
S1 Unaided	38%	12%	75%
S1 with SC	5%	8%	93.5%
Δ	-33%	-4%	+18.5%

3.2 Test 2: Robustness of the Operator Model

One hypothesis regarding human performance modeling is that differences between people are large enough to require individual, or idiographic, models. To investigate this possibility, we tested the performance of operators with the Stochastic Controller (SC) when the SC contained a model of a different person. If human behavior is sufficiently similar in this venue, then overall performance with the SC should still benefit from the ‘mis-matched’ model compared to unaided performance. Previous analyses has shown that operators react to changes in Ground Sample Distance (GSD), Aspect Angle (AA), and Dwell Time (DT). These had the most impact in prior tests, therefore we expected that the operator error model in the SC may be able to generalize to other similar (suitably trained and selected) operators.

3.2.1 Method.

3.2.1.1 Subjects.

Two members of the research team volunteered to participate as test subjects (S2 and S3). Both subjects were male with normal or corrected to normal vision and ages ranged from 24 to 28 years (mean = 26). Both subjects had previous experience in psychological research and had full knowledge of the research goals and task.

3.2.1.2 Task and Training.

This test used the same test apparatus and task from the operator S1 test. In addition, the SC used the model of operator S1 for both subjects. Both subjects were trained in the task, one subject trained for 2 runs and the other trained for 5 runs, with sufficient performance for participation set to PC ≥ 70% correct. A total of 17 runs per subject were conducted, resulting in 340 trials per subject for analysis.

3.2.2 Results and Discussion.

Table 3.3 shows the results from these two subjects in comparison to operator S1’s results from the SC assessment from Test 1. Mean confidence values for the operator with the SC were calculated by replacing the first look confidence rating with the second look confidence rating for revisited targets. Data showed a large degree of similarity across operators. Unaided performance for the different operators was similar in the task, as well as the direction and magnitude of the benefit of the SC. These data suggest that an operator error model for AA, GSD, DT, and Inter-Event Time (IET) may be robust and generalizable to different subjects that are similarly trained and selected. In other words, these data did not support the hypothesis that operators needed an individualized representation of their performance in order to benefit from the SC.

Table 3.3: Results of Robustness Test

	FA	MD	PC	*d'*	Mean Confidence
S1 Unaided	38%	12%	75%	1.48	1.8
S1 with SC	5%	8%	93.5%	2.82	3.3
Δ	-33%	-4%	+18.5%	+1.52	+1.5
S2 Unaided	31%	23%	73%	1.25	2.2
S2 with SC	7%	11%	91%	2.69	4.0
Δ	-24%	-12%	+18%	+1.44	+1.8
S3 Unaided	28%	16%	78%	1.57	2.2
S3 with SC	5%	11%	92%	2.81	3.5
Δ	-23%	-5%	+14%	+1.24	+1.3

IV. Tests at Target Density of 5%

The proportion of targets in an environment (target density) can change from one mission to the next. Presumably, the majority of current missions have many Points of Interest (POIs) to view, few of which are targets, though this may not be true for all missions. Therefore, we thought it important to study the performance of the operator and of the operator error model within the Stochastic Controller (SC) under multiple environments, some with equal number of targets and non-targets (as described above) and some in which targets are relatively rare. This chapter describes work to understand how operator performance may change in a low density target environment and test the operator error model's robustness to different target density.

4.1 Test 3: Predicting Operator Performance at Target Density of 5%

The concept of the Stochastic Controller (SC) is to plan revisits to points of interest when the human operator is likely to make confusions/errors under the conditions/state of the visit. The SC requires some representation of the operator's performance under different conditions in order to select the conditions of relatively poor performance (bad states) and avoid unnecessary expenditures of resources, such as fuel, on revisiting good states. The representation, or operator error model, of the human's performance within the SC is a confusion matrix. The confusion matrix represents the human's tendencies to the machine in order for the machine to compensate for difficult and low performance situations for the human and simultaneously take advantage of and conserve resources during easy and high performance situations. In addition to the confusion matrix, this research explored an additional human behavior model using signal detection theory. We used the data from the 50% density tests to construct a signal detection model of the operator. Then we used the signal detection model to attempt to predict how performance would change in a low density task.

4.1.1 Method.

4.1.1.1 Data Used in Constructing the Model.

The model was constructed using the data from Test 1 (120 runs with operator S1). That test manipulated four variables: Aspect Angle (AA), Ground Sample Distance (GSD), Dwell Time (DT), and Inter-Event Time (IET). As mentioned earlier, but distinct from many signal detection studies, AA, GSD, and DT were randomly sampled from uniform distributions within their minimum and maximum allowable values (Table 2.1). IET was not controlled but measured post hoc, and its distribution was non-uniform with a positive bias. Typically, signal detection studies typically present the same stimulus multiple times over the course of an experiment. The methodology used in the present study did not have such repetition, so within the analysis the stimuli were grouped and the grouping strategies are described below (median-split was used most often). Another distinction from many studies of note is that the blocks contained trials of randomly varied independent variables (i.e. “first looks”) mixed with trials that were revisits of points of interest with a flight trajectory to maximize performance (i.e. “second looks”), which were not used in creating the signal detection model.

According to signal detection theory, when targets are very rare, operator behavior changes as a result of changes in decision bias and not from changes in sensitivity. We used the signal detection analysis of the first look data at the 50% density to create a model of the operator’s sensitivity for high and low levels for each of the four variables (Figure 3.2), and changes in bias were estimated using past findings. The target density of the environment, also known as the a priori probability of a target, was set to 0.05 for this modeling work. The theoretically optimal bias for maximizing percent correct is 19 ($P(NT)/P(T)$), however research has shown that humans use suboptimal biases for a priori probabilities less than 0.30 or greater than 0.70 [3, 12]. No published data were found for 5% density, so we took data from 10% density [3, 12] and extrapolated to arrive at an expected bias of $\beta = 3$ to use in the model.

4.1.1.2 Prediction Procedure.

The synthesized responses were generated by: 1) determining the trial type (p(target) = .05, p(non-target)=.95), 2) randomly draw values for the GSD, AA, DT, and IET (a "state") from distributions measured in the first experiment, 3) look up the sensitivity for that state in the operator model, 4) calculate the decision criterion corresponding to a bias of 3, 5) randomly draw an event from the target or non-target distribution (depending on trial type), and 6) synthesize the response (i.e., target response if the event is above the decision criterion and non-target response if the event is below the decision criterion). A total of 50,000 trials were synthesized. Examples of synthesized responses showing one of each of the four decision outcomes are found in Figure 4.1. The diamond symbol near the abscissa represents the event. A blue diamond corresponds to a non-target event drawn from the non-target distribution and a red diamond corresponds to a target event drawn from the target distribution. Also note how changes in *d*' result in changes in overlap between the target and non-target distributions.

4.2 Results and Discussion

Model predictions of operator performance in the 5% density are shown in Table 4.1. These predictions are presented next to S1's data from Test 1 for comparison. (Comparisons between model predictions and operator performance at 5% density to validate the model are shown below in Table 4.3). Note the increase in PC and the decrease in hit rate and FA rate that are expected as a result of the change in density.

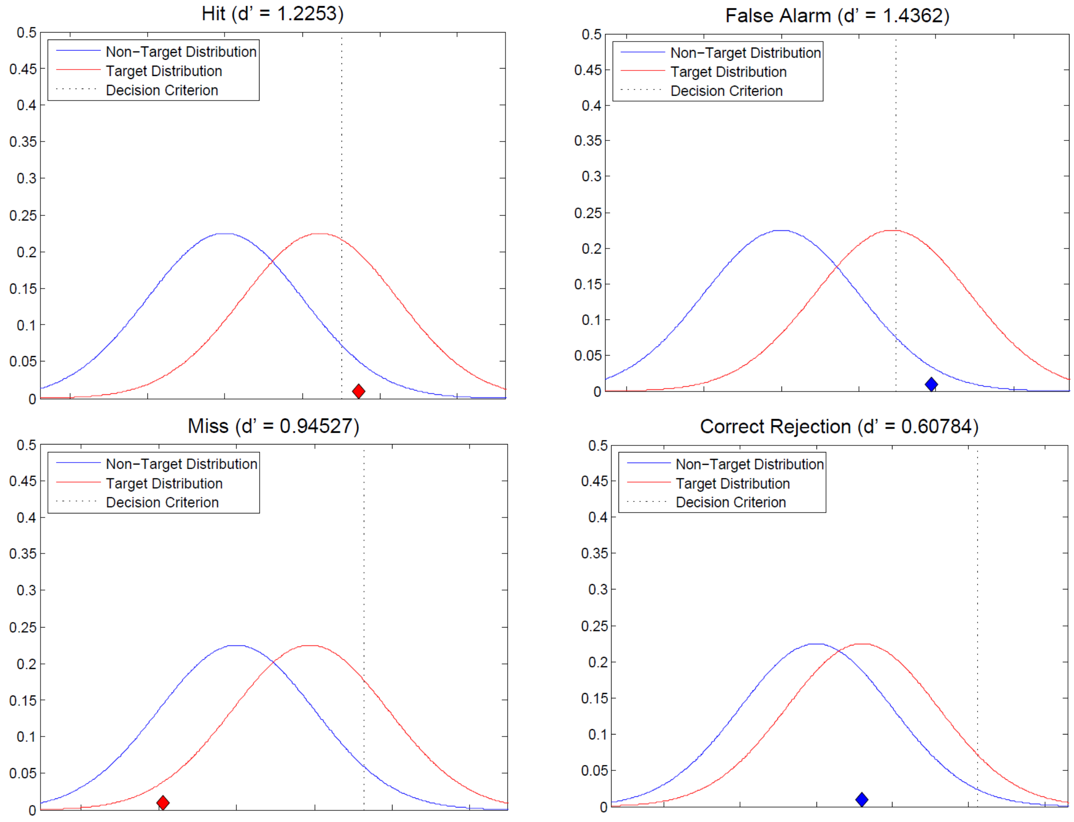


Figure 4.1: Examples of Synthesized Responses in Low Target Density

4.3 Operator Test at Target Density of 5%

We tested operator performance with and without the SC at a target density of 5%. The objective of the test was twofold: 1) to determine the robustness of the SC with a 50% density operator error model in an environment where targets were rare, which will be called a “low density” scenario, and 2) to validate the predictions made by the signal detection model of operator behavior. The task was modified from initial tests so that there was only one target per run of 20 Points of Interest (POIs) (5% target density). This change was made in all 4 scenarios used in earlier tests. To try to minimize predictability of the

Table 4.1: Predictions using the Signal Detection Theory Model Compared to Operator Performance

	5% Predictions	S1's 50% Performance
PC	92.84%	74%
Hit Rate	42.59%	83%
FA Rate	4.50%	35%
d'	1.51	1.34
β	4.14	0.68

target and ensure the operator monitored all 4 Unmanned Aerial Vehicle (UAV) streams, each scenario had a different UAV which flew over the target.

4.4 Method

4.4.1 Subjects.

The same test subject from Test 1 was used (operator S1). The test subject was not informed of the change in target density.

4.4.2 Task.

The same procedure from Test 1 was used including the operator error model developed under 50% target density, with the exception that each run had 1 target and 19 non-targets. The change in density was input to the SC, as mentioned above. The operator performed 20 runs, resulting in 380 non-target trials and 20 target trials.

4.5 Results and Discussion

As expected, operator S1 detected the reduction in the number of targets per scenario very quickly. When asked how many targets each run contained, operator S1 estimated 1 or 2, which is correct, assuming that on some trials the target Point of Interest (POI) may have been revisited. The a posteriori probability of target affected the operator's conscious strategy (i.e., bias). She responded Non-Target unless she had very high confidence (a rating of 5) that the object was a Target.

The results from the low density test were analyzed using the 1st 10 runs, the 2nd 10 runs, and finally all 20 runs together. The data were split to investigate the possibility of learning or a change in strategy/decision criterion during the course of data collection. The results summary are shown in Table 4.2. In the Table,

PCtotal = (#NonTarget Hits + #Target Hits)/(#NonTargets + #Targets)

PCavg = (Ptr + Pftr)/2 = (#Target hits/#targets + #NonTarget hits/#NonTargets)/2

Table 4.2: Results of Low Density Test

		FA	MD	PCavg	PCtotal	*d*'
All 20 runs	S1 Unaided	1.8%	60%	69%	95.25%	1.84
	S1 with the SC	0%	25%	87.5%	98.75%	3.33
	Δ	-1.8%	-35%	+18.5%	+3.5%	+1.48
1st 10 runs	S1 Unaided	2.6%	50%	74%	95%	1.94
	S1 with the SC	0%	30%	85%	98.5%	3.18
	Δ	-2.6%	20%	+11%	+3.5%	+1.23
2nd 10 runs	S1 Unaided	1.1%	70%	65%	95.5%	1.77
	S1 with the SC	0%	20%	90%	99%	3.50
	Δ	-1.1%	-50 %	+25%	+3.5%	+1.73

In comparison to the operator's performance at 50% density, there was a dramatic reduction in errors for the 5% density. Total Percent Correct (PC) went from 74% at 50% density to 95.5% at 5% density. These differences are most likely due to the change in density, the primary change between tests. There was no change in the distinguishability of the features (Y versus V) or in the range of states at which the objects were viewed.

From the perspective of signal detection theory, the operator's decision threshold (or bias/strategy) changed between the 50% and 5% scenarios ($\beta = 0.53$ and 8.73, respectively). The question is: what value does knowledge of the density have on predicted performance? If the operator responds non-target to everything (extremely conservative bias), then FA=0, MD=1, and PCtotal=95%. The operator's observed performance for the 5% density is also PCtotal=95%. The implication, using this criterion, is that the observations are no better than just saying no. An alternative policy is to decide by throwing a weighted die. In this case, the die would only say Target 5% of the time. Note: 5% is the true density—there is no density error. The computation now is PCtotal = $(.05*.05+.95*.95)*100$ = 90.5%. The metric PCavg=50% for "always say no" as well as the loaded die. The metric Information Gain (IG)=0 for both the "always say no" and the loaded die. This will be discussed further below in Appendix H on Information Gain. The metric PCtotal is not very sensitive to the operator observations at low density.

4.5.1 Estimator Results.

The Stochastic Controller and Estimator for these runs had the 50% density derived operator error model *LEAPmodel0920.xml*, as well as the true information that the target density is 5%. Of the 9 errors in the 2nd 10 runs, the Estimator caught the 2 False Alarm (FA)s and 5 of the 7 Missed Detection (MD)s. If you use the density invariant PCavg metric, then PCavg goes from 65% to 90%. The size of the performance improvement with this metric appears to be consistent with the 50% density PCtotal=PCavg going from 74% for the operator to 93% with the operator and SC.

Without conducting additional tests, it was not possible to determine what the performance improvement would be if there were no operator modeling error. Obviously, the reduction in average operator FA rate from 35% to 1% was huge. The SC, with the erroneous operator model, was predicting an average of 2 FAs and possibly a Target for each UAV. This would give an average of 3 revisits needed for each UAV, so that all of the objects that the operator declared a Target can be looked at again, so each UAV would have just enough reserves. However, operators did not make target responses that frequently. In general, this meant that the controller husbanded the reserves, so there is little chance of running out before all 5 of the objects are visited. This appeared to be born out by the actual SC performance, where the last 3 objects were revisited 80% of the time and the first 2 objects were revisited 20% of the time. By design, all of the reserves are used in every run.

The modeling error has little consequence for FAs because, at low density, the SC would revisit essentially any object that the operator says is a target. The only exception would be if the operator FA rate were essentially 0 at the specified state. This is what happened with the present data—the SC revisited all the target declared objects and the Estimator correctly declared the Non-Targets as FAs. So, the operator modeling error has shown little effect on the Estimator ability to correctly classify FAs.

At low density, the situation is different with MDs. In general, a Non-Target declaration by the operator results in the Estimator P(T)<density or P(T)≪density, so there is little expected Information Gain from revisiting the object. However, there may be a combination of FA and MD at that state that could result in a revisit decision. An example would be if the operator is a contra-indicator—if the operator says it's not a Target and there is a significant probability that it is (which was not the case for any operator we tested). If this is in the model, the SC can pick it up and exploit it correctly. So, baring these cases, the SC, because of the low IG, will not normally revisit an object that the operator has declared a Non-Target.

However, if there are no Target declarations by the operator, the SC will still revisit up to 3 NonTargets to use all the 3 revisit reserve. Essentially, nearly all of the reduction in MD, for example from 7/10 to 3/10, is due to this effect. The operator modeling error biases the MD revisit to the last 3 objects in the 5 object inspection sequence. If all the actual targets are in the 3, 4, or 5 position, then all the MDs will be revisited and the effective MD rate will be 0 (given sufficient reserves). In actuality, across the 4 scenarios Targets appeared in the 1st, 2nd, 3rd, and 5th positions, making the scenarios slightly biased toward the beginning of the sequence. If the situation were reversed, the effective reduction in MD would be somewhat greater. Also, a reduction in operator MD modeling error should result in more objects being revisited at the 1 and 2 positions. The point here is that it is much more difficult to reduce the MD rate than it is to reduce the FA rate at low density. Presumably, this would be reversed in a high density environment.

4.5.2 Validation of the Signal Detection Theory Model.

The data from operator S1 in low density are presented alongside the predicted operator performance from the signal detection theory model in Table 4.3. The measured bias of the operator was higher than expected and in disagreement with the values taken from the literature. The false alarm rate and the hit rate were lower than predicted, presumably due to the difference in bias. There was some possibility that the difference in bias was due to sampling error from the small sample size (20 of the 400 trials presented a target). In addition, the bias measured from the synthesized data was higher than the theoretical bias of 3.0 used to generate the data. To investigate the possibility of sampling error as the cause of the bias increase in the synthesized data, synthesized data was generated for 100 repetitions of 400 trials. This repetition of 400 trials allowed an investigation of the expected variability in bias due to the sample size collected from the operator. Furthermore, these tests were run with many different theoretical bias values and found that variability in measured bias increases quickly (Figure 4.2). For theoretical biases of 3.75 and higher, the measured bias of 8.45 falls within 2 standard deviations. Most likely, there still is a difference between the theoretical and measured bias, but it is not likely that the difference is as large as the first comparison suggested. The differences in *d*' may be due to the operator's increased ability to learn the 4 experimental scenarios because fewer targets were present (i.e., “game” the task by leveraging reduced uncertainty). During a debrief interview, the operator indicated the ability to partially learn a sequence between which UAV would visit the target in a scenario.

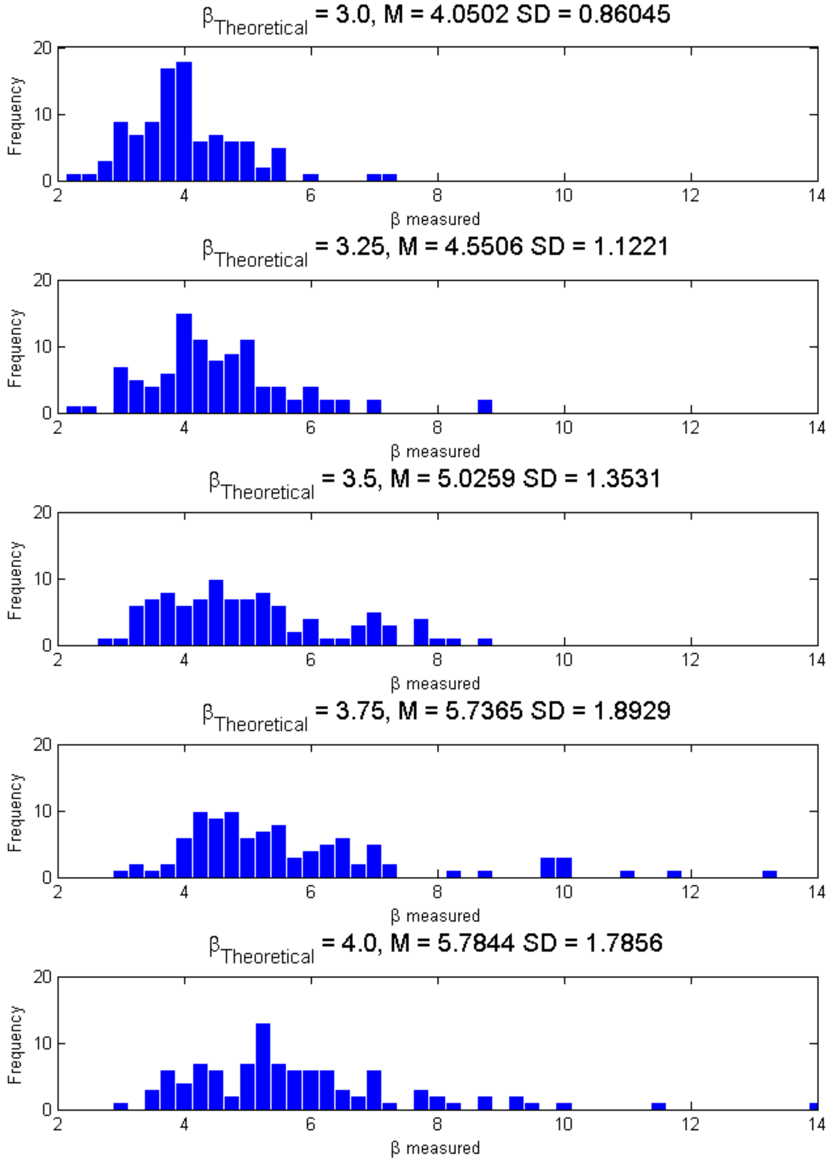
Table 4.3: Validation of the Signal Detection Theory Model in 5% Density

	Model Predictions	S1’s Performance
PC	92.84%	95.24%
Hit Rate	42.59%	40.00%
FA Rate	4.50%	1.85%
d’	1.51	1.83
β	4.14	8.54

4.5.3 Issues.

Deterministic Density and Scenarios

For each scenario, only 1 of the 20 POIs the operator looked at was a target. From the debrief, it was clear the operator had determined the number of Targets. The debrief also revealed that the operator knew there were 4 distinct scenarios, though the operator was never told this. It is possible that some of these results are confounded with the operator’s ability to exploit predictability of targets and scenarios, which would have been the expectation that the targets appeared in a particular UAV’s sensor feed in one run, followed by appearing in another UAV’s sensor feed in another run (e.g., UAV 2 had the target in the run after UAV 4 had it). This would reduce the decisions made by the operator and perhaps allow him/her to ignore a UAV’s feed on the run after it had seen the target. A possible solution for future work would be to increase the number of scenarios and randomize the order of scenarios so that target appearances were not related to UAVs and never followed in a pattern. Additionally, the number of targets within a scenario could



$\beta_{Theoretical}$ = 3.0, M = 4.0502 SD = 0.86045

$\beta_{Theoretical}$ = 3.25, M = 4.5506 SD = 1.1221

$\beta_{Theoretical}$ = 3.5, M = 5.0259 SD = 1.3531

$\beta_{Theoretical}$ = 3.75, M = 5.7365 SD = 1.8929

$\beta_{Theoretical}$ = 4.0, M = 5.7844 SD = 1.7856

Figure 4.2: Inflation of Synthesized Bias Values

be varied to achieve 5% density across the scenarios but have 0, 1, or 2 Targets POIs per scenario. Given the high MD rate, it is clear that the operator did not leverage this predictability to its full extent.

Low Target Samples

In the low density scenarios tests, there were only 20 Target samples. If one says the 1st 10 runs were training, then there were only 10 samples. This was too low to be able to make a reliable estimate of the MD rate. To get 100 Target samples, one would have to make 80 more runs. This was impractical, and it shows the limitations in extracting operator data at low densities. Also, since the simulation was stochastic, the operator may have seen more bad Target states than good Target states in this small sample, further biasing the results.

Evaluation Metric

A single number is a useful metric to compare performance results. We have used PCtotal, PCavg, and IG primarily. PCtotal is probably the most used, but there are issues in applying virtually any metric at low density. As one can see in this section, as density → 0, then PCtotal → 100%—just by doing nothing. This does not provide much insight into comparing alternatives.

Modeling Error

Given the impracticality of obtaining valid low density operator data, there is a robustness issue in using a high density operator model in a low density scenario. If one looks at the FA rate, then the policy pursued by the operator appears to be quite different from FA=35% at 50% density to FA=1% at 5% density. Using the model to predict performance at low density would incur significant error. A valid signal detection theory model of the operator could be used to develop the operator error model with less modeling error.

4.5.4 Summary.

The most substantial difference between the operator performance at 5% density and 50% density was the radical reduction in the number of classification errors (from 52 to 9). We concluded that the operator used a different decision threshold or classification strategy to make target/non-target responses. The strategy is indeed, density dependent. From a Signal Detection Theory (SDT) point of view, the operator is extremely conservative—not willing to tolerate any FAs to achieve detections (FA=1%, MD=70%). Presumably, the operator was maximizing PCtotal or minimizing the total misclassifications (in the presence of equal values, costs/payoffs for Hits, MDs, and FAs). An interesting comparison can be made to a "always say NonTarget" policy, which yields 10 errors (in 10 runs). This compared closely with the 9 errors in the 10 runs. One could say that the operator observations at low density provide little additional information or value.

The behavior of the Stochastic Controller can be usefully compared to a 2-stage screen or a 2-stage filter. The 1st stage of the screen applies a low cost, low fidelity test to the population. Those objects passing through the 1st screen are passed to the 2nd screen, which consists of a high cost, high fidelity test. Those objects passing through the 2nd screen are declared Targets (though more processing could be done). Due to the high cost, only a limited number of high fidelity tests can be done. The objective is to design the 2 screens (SC and Estimator) so that incorrect classifications are minimized.

The demonstrated performance of the SC and Estimator has resulted in FA=0%, MD=20%, and PCtotal=99%. As can be seen here, all of the FAs have been screened out and 80% of the true Targets have been detected. This in spite of the error incurred by modeling the operator using 50% density data. It is impractical to model the operator using low density data because of the amount of data required. In this instance, the SC and Estimator have been shown to be robust to operator modeling error. This may not be true in general.

An alternate configuration of the 2-stage screen uses additional confidence [1–5] information. A 1 signifies a no confidence in the operator's [Target/NonTarget] declaration and a 5 a very high confidence. The objects passing through the 1st stage screen have confidence of [1,2]. This reduced set is revisited at the same high fidelity state as the SC revisit state(s). This revisit state has few, if any, classification errors. Whatever the operator says [Target/NonTarget] will be the object declaration. This strategy performs surprisingly well and does not require an operator model that is a function of state.

For the confidence factor strategy to perform well, it is important that the total number of [1,2] confidence levels does not exceed the available reserve per UAV. No effort has been made to prioritize the sequence, so it will just be truncated. An important point is that any Target that the operator has falsely declared a NonTarget will have to have a 1 or2 confidence level to pass thru the 1st stage screen to the 2nd stage to have a chance to be reclassified as a true target. This is somewhat different than the SC strategy, which is to effectively pass all Target declarations to the 2nd state as well as to sample from the 1st stage population of NonTarget declarations to fill out the revisit reserve. This has the potential to reduce FAs to 0 while reducing MDs somewhat.

The final point here is that it is much more difficult to reduce the MD rate than it is to reduce the FA rate in a low density scenario.

V. Summary and General Discussion

5.1 Integrated Operator Model and SC at 50% density

This line of research began almost 10 years ago as an extension of stochastic optimal control. The thrust of the controller is to determine the best action to take now, to achieve an overall goal in the future. The extension is to include the operator in the controller design. The earliest effort looked at screening the Automatic Target Recognition (ATR) results to determine which engagements were expected to provide the most information for the operator to review. The patrol problem addressed which alert to view next based on an arrival rate and the operator error rate. The sequential inspection problem involves determining which objects to revisit to maximize the probability of correct classification.

These earlier works utilized a simple generic operator model, such as FA=MD=20%. The benefit of such a simple model is that it introduces additional uncertainty into the controller design. The fact that one has to allow, in the design, for the operator classification to be wrong, may be just as valuable as the accuracy of the operator error model itself. In the lower density scenarios particularly, the controller will tend to revisit any object the operator initially says is a target. During field tests, the operator would tend to ask the test director why the Unmanned Aerial Vehicle (UAV) is revisiting the object after the operator has already said it is a target. The frequently uncomfortable answer is: “you could be wrong.” This points out the issue of operator acceptance of a decision system that can be viewed as second guessing him.

One can also quote rules that state that there is insufficient evidence to convict the object as a target. The Bayesian approach requires strong evidence (low error) to convict if the prior is low (low probability of a target). However, the controller/estimator can be wrong as well—it all depends on the fidelity of the models, particularly the operator error model. Since the operator's errors were seen to vary widely, quantizing the errors as a function of the most significant variables seemed a reasonable choice.

Modeling the operator was not originally part of the Stochastic Controller (SC) research direction. It came as a consequence of accounting for the operator in the controller design. The desire for better (more predictable) performance inevitably leads to the need for better models. Modeling the operator is problematic because: 1) operators vary widely; 2) probabilistic models over many conditions requires extensive empirical data; and 3) low target density results in extremely low target data. The LEAP II effort addressed the 1st issue by aggregating many operators into a single average operator model. The consequences are that if a particular operator is far from the mean, then predictions are poor. The present Integrated Human Behavior Modeling and Stochastic Control (IHBMSC) effort avoids this problem by tailoring the model to the specific operator. The 2nd issue is addressed by having the model be a function of dynamic variables only, and samples are restricted to the high/low ranges. This allowed the number of states to scale by 2^n. Both LEAP II and IHBMSC took this approach. The 3rd issue is addressed by modeling the operator at 50% density and using the incorrect model at the lower density. Again, both LEAP II and IHBMSC took this approach, where the IHBMSC project in particular was found to be quite robust to the modeling error.

An issue that absorbed considerable time in both LEAP II and IHBMSC was the False Alarm (FA) rate. This stems from basic issues in detection as well as screening techniques, particularly at low density. The SC, being a 2-stage screening technique, is more adept at reducing false positives than it is at reducing false negatives. Signal Detection Theory (SDT) says that for a specific signal/noise ratio, the detection rate can only be raised by lowering the decision threshold, which raises the FA rate. While a threshold can easily be lowered, say in ATR, it is not at all clear how to lower the threshold of a human operator.

In addition, as the density decreases, the operator FA rate seems to naturally decrease. This behavior has been observed by Operator S1 as well as by Operator S0.

Obviously, if there are no operator FAs, then there are no FAs for the SC to reduce. The initial conclusion was that the scenarios were too easy. The objective then was to make the discrimination problem more difficult by making the Targets and NonTargets more similar. This was done in both LEAP II and IHBMSC by changing the states. The result was LEAP II has average operator FA=32% and IHBMSC has average operator FA=35%, both at 50% density. This is the point where the IHBMSC operator modeling exercise begins.

The decision was made not to separate the estimation process from the control process. While this could have been a problem, it turns out that the estimation and control are reasonably independent. The model data is collected at the same time the controller performance data is collected. The model is updated periodically, while the controller is evaluated using the previous model version. This allows an early assessment of performance before all the model data is collected—thus preventing any nasty surprises. While this may not be the best approach in general, it seems to work well in this case.

One consequence of the simultaneous estimation and control is that the modeling became fairly straightforward. The number of samples at each of the states was reasonably uniform. This meant a 16 state table could be constructed without resorting to any complicated parametric modeling. A 256 state model was then constructed to interpolate between the 16 states.

The final operator performance model for Operator S1 at 50% target density was FA=35%, MD=17%, and consequently, PC=74%—from data taken over 120 runs. The final result of the SC (with final operator S1 model) + operator S1 observations + Estimator during the ten 50% density evaluation runs was FA=5%, MD=8%, and PC=93.5%. The FA rate was cut by a factor of 7, and the MD rate was cut by a factor of 2. These are average results, where individual runs can be better or worse. It shows that operator performance can vary widely, for many reasons, not just at different states. It shows that a decision aid, in conjunction with the operator, can effect a radical improvement in system performance. It shows that the decision aid is not second guessing the operator, but complementing the operator.

5.1.1 Comparing Operator decisions to Estimator decisions.

First, take the 120 runs by Operator S1 that are used to build the operator error model 0920 and extract the operator 1st look responses where there were no 2nd looks. Second, to this add the 2nd look operator responses for the cases where there was a revisit. For this list, the overall decision making performance of the operator is PC=92.03%. The overall decision making performance of the Estimator for the same 120 runs is essentially the same. The component FA and Missed Detection (MD) rates are essentially the same as well.

However, the Operator and Estimator equivalence is not a generic result. It is, perhaps, an artifact of the “clean” nature of the final scenario definition, plus the consistently good performance of the operator. The operator essentially had no errors for the 2nd look observations (either in the model data or the evaluations runs). This is because the SC always had the operator observe the object from the best available state (reflected in the operator error model). Plus, there was no variation in the revisit state (hence clean). In hindsight, this may not have been the best thing to do. Some variation would have been normal. The Estimator correctly weighted the 2 observations together probabilistically using Bayes rule, with no tuning of the threshold at all, and consistently made the correct decision. Operator S1 is a high performing operator and the SC is intended to aid the operator. But, less aiding is required if the operator is very good in the first place. This means that one is less reliant on the SC (which includes the Estimator) and heuristics/simplifications could be okay. Also, a less proficient operator would benefit more from the SC decision aiding.

The operator Confidence Level (CL) observations constitute a “simple” confusion matrix. Given the present clean scenario, however, the CL model can be a useful approximation—when used in conjunction with simple heuristic rules. The SC, in contrast, is generic and mathematically rigorous, yielding the optimal estimate as well as determining which objects to revisit using the available resources. If a confusion matrix is desired to be a function of more variables (greater fidelity), then the operator data taking, in particular, can become intractable—especially at low density. The CL model is not (at least here) an explicit function of state and therefore scales better, but still requires operator data to correlate CL with performance. However, for more ambiguous scenarios, the simple CL model could lead to cumbersome decision rules and resulting less satisfactory performance. Complicating factors include: low density; the operator is not monotonic, meaning the performance is not constantly increasing with CL; the operator is a counter indicator; the conditions are much more ambiguous and it’s not clear what to do; there is no stand-out “best” state; the operator is inconsistent and a lower performer generally; and finally, there may be a need to combine multiple (similar) observations.

5.2 Integrated Operator Model and SC at 5% density

Empirical modeling for low density targets, and low probability of occurrance events in general, present difficult issues. It is impractical to generate the requisite data, especially for an operator in the loop simulation. This is important, because the experimental evidence in this project has shown that the operator behavior is radically different between the 50% density scenarios and the 5% density scenarios. On average, the operator will make 52 errors at 50% density and only 9 errors at 5% density. The operator is heavily conservatively biased and will only declare an object a Target if the confidence level is high (a 5). This results in the operator FA being near 0, but the operator MD can be as high as 70%. There is not enough data to be more accurate with the MD rate.

Experimental tests were performed by Operator S1 using the SC with the Operator S1 50% derived error model on 5% density test cases. This was done to test how sensitive the SC results were with the modeling error. Surprisingly, the SC worked extremely well, with FA=0 and MD=20%. The SC eliminated the (few) FAs, as expected, but also made a significant improvement in the detection. This is primarily due to the number of revisits (60%) enabled in the design and the very low error at the revisit state. Also, the SC has knowledge of the 5% density—the operator has knowledge of the 5% density as well. In summary, the SC can overcome some of the extreme operator conservancy to increase the overall target detection rate.

Since operator confidence level correlates quite well with performance, there is the possibility of being able to implement a heuristic control policy that does not depend on an explicit operator error model. Essentially, the heuristic controller will revisit when the operator has the least confidence (a ‘1’), plus possibly a few ‘2’s to fill out the allowable revisits. The final determination is whatever the operator said on the revisit. This has the promise to perform reasonably well. If the operator performance does not correlate well with his confidence level, then this would complicate a heuristic policy. This is not an issue with the SC because the operator error model encodes the operator’s true performance, not the operator’s perceived performance, which is what is encoded in the confidence level.

Alternatively, SDT has the promise of enabling the development of a 16 state operator error model when there is insufficient data to estimate the MD rate—which is the case at low densities. There is sufficient data to estimate the FA rate at the 16 states, however. SDT was shown to be able to leverage existing or ongoing research to estimate operator bias at low densities. Thus yielding an estimate of MD rate at all 16 states which can be incorporated directly into the SC without the need for heuristics.

The utility of the Percent Correct (PC) performance criterion at 5% density is questionable. As was discussed before, if the operator always said NonTarget, then

PC=95%. This is not a sensitive enough criterion to allow one to evaluate different approaches. Intuitively, if the operator policy is to do nothing (or always says the same thing), then the value of that policy should be 0. This is indeed the result if the Information Gain (IG) criterion is used to evaluate the "do nothing" policy. If the operator should flip a target density weighted coin, then the IG criterion also says the value of the policy is 0. The IG criterion is a good candidate for use on low probability of event decision problems.

5.3 Recommendations

Operator Modeling

There was significant operator modeling error in using the 50% density developed model for a 5% density problem. For identical scenarios (10 runs, 200 objects total), which differed only in density, the 50% density model would predict 69 operator errors, while in fact there were only 9 operator errors observed on average. Obviously, the operator error model is a significant function of density. The difficulty lies in empirically modeling the operator, in particular Missed Detection (MD) rates at low densities. At 1% density, one would need 9,900 Non-Target samples just to exercise the operator at 100 Target samples suitable for estimating a MD rate across all states—more samples would be required to sample MD rates for each of 16 state bins. Clearly this is expensive. A functional model of the operator is needed. From Signal Detection Theory (SDT), it can be seen that Target/Non-Target distinguishability is independent from bias. Given that this analogy holds for operators, then the problem is to analyze how and why the operator's bias shifts just from the perceived density changes.

Prior

The Stochastic Controller (SC), Information Gain (IG), and the Estimator are all a strong function of the prior information on the target density. Knowledge of the Target density is assumed known, or at least a good estimate of the density is available. Analysis of the sensitivity of the results to small and large errors in density should be done.

Realistically, trying to estimate the prior target density may be just as difficult as trying to estimate the operator errors.

Operator Bias

A limited attempt was made to have Operator B bias her decision to be more liberal—in effect to drive up the False Alarm (FA) rate, which in turn would drive up the detection/hit rate. The results of these efforts were erratic, and it was decided not to pursue this line of investigation. However, pursuing this line may be productive in increasing detection at low density. This would counter what appears to be the operator's natural tendency to lower FA at lower density. This plays into the strength of a 2-stage filter (of which the SC is an example) to significantly reduce the FA rate with much less reduction effect on the MD rate.

Reserve and Threshold

One objective of the SC design was not to have any "tuning" parameters. All parameters were to be a function of the scenario definition. The reserve and the decision threshold may be exceptions to this guideline. The Unmanned Aerial Vehicle (UAV) revisit capability (reserve) is based on the expected number of operator errors over the expected states visited. Following is a discussion of the number of revisits needed (hence reserve). As previously discussed, there were 52 errors by the operator in 10 runs at 50% density, giving 5.2 errors per run. However, this is an average, so use factor of 2 to get 10.4 errors per run. Two revisits per UAV gives 8 total, and 3 revisits per UAV gives 12 total. The 12 revisits were used for all of the SC calculations. However, this may not be optimal and the sensitivity of the results to the reserve should be studied.

The default decision threshold was set to the prior density. When the computed P(T) was greater than the density, the Estimator declared the object a Target. This may not be the optimum choice, so the determination of the threshold needs to be studied.

Different Features

The operator had to discriminate between a **V** and a **Y** in the present effort. There are many other choices of interest, such as a subject with a weapon versus a subject with another object that looks similar to a weapon or a weaponized vehicle versus non-weaponized vehicle with similar cargo. In addition, the vehicles/subjects could be moving, which presents a challenging planning and tracking problem. Each of these variations presents their own challenging and unique aspects.

Operator Cueing

In these tests, the operator was cued to the presence of an object by illuminating the viewing window. Also, the operator observation of the object was input using a Target or a Non-Target button. The presentation to the operator should be studied to optimize the operator performance. Additional information could be relayed from the operator by using more buttons to signify more classes—such as confidence level.

Confidence Factor

A sliding scale could be used to signify confidence in the observation. The confidence information could then be used to design a controller to detect and reclassify errors. Additionally, the alternative revisit strategy using confidence ratings (Appendix K), which does not use information theory, could be implemented and tested with operators for a comparison to the SC.

Online Modeling

A high fidelity mission simulation could be used to build an operator error model. The SC would not have access to ground truth so supervised learning would have to be used to develop the Estimator's classification strategy. This supervised learning approach could incrementally improve the model over a larger range of conditions. The model could be directly incorporated into the SC. The key limitation is the amount of data needed for a model which is a function of many variables. However, all mission training/practice time could produce model data.

Adaptive Interface

Just as the HBM-informed stochastic controller modified the mission plan and resulting aircraft routes, the controller could be used to manage adaptive user interface controls and displays. For example, the controller could monitor and assess the target acquisition conditions, and where needed, manipulate the video presentations. The video capture from the first look could be recorded, queued up, replayed, slowed down, and/or possibly digitally zoomed in. A prototype design is currently being developed and a human-in-the-loop evaluation is planned to assess the effectiveness of the stochastic controller and adaptive interface integration.

Knowledge of Intent & Level of Automation

Further research is needed on whether or not the system should provide the operator insight into what the stochastic controller is calculating, along with the level of automation appropriate (e.g., management by consent, management by exception, full auto) for enacting changes to the mission plan or the video imagery presentation.

Proactive Re-Planning

In addition to referencing the human performance model for the stochastic controller's re-visit calculations, the model could also serve as a reference for determining if and how adjustments should be made to the flight paths before the vehicles fly over their assigned target locations. For example, if an inflight disturbance, such as strong winds, was causing the vehicle to drift off course or have a significantly faster ground speed than planned, the model could be referenced to project and assess the revisit conditions, and where out of acceptable regions, the route(s) could be modified to improve the target acquisition situation.

Level of Automation Comparison Study

To assess the value of a stochastic control decision support system that re-plans routes automatically, a formal human-in-the-loop study is needed to compare target acquisition performance between a system with a stochastic controller and one without that still incorporates revisits (such as operator manually-selected revisits).

Variables/States

These tests studied Ground Sample Distance (GSD), Aspect Angle (AA), Dwell Time (DT), and Inter-Event Time (IET). There are more variables of the object inspection task (both perceptual conditions and flight path parameters) that were not included but impact target detection performance. (additional list of perceptual things from USAFSAM white paper). Future work could expand the variables studied and possibly controlled for in dynamic mission planning.

Inter-Event Time

IET was not controlled in this study, but measured post hoc. Moreover, the data from Operator S1 showed IET had little effect on performance, contrary to expectations about event rate, workload, and previous findings in the literature. Further investigation into this phenomenon should be conducted, and would require control/explicit sampling of IETs.

Model Robustness to Different Operators

A preliminary test was conducted that showed that the operator error model was robust to different operators, and suggested idiographic performance models may not be needed. This test was small in data collection and only tested one operator error model under this specific set of variables. A “gold standard” test would be to create a complete operator error model for additional test subjects and then quantitatively examine the differences between their models and test their performance with each other’s ‘mismatched’ models. Perhaps for variables of GSD, AA, DT, and IET operators aren’t similar when studied in this more thorough fashion. Moreover, other variables of interest mentioned above may show additional differences between operators (e.g., contrast sensitivity is known to vary between people and result in differences in performance on target ID tasks).

Bibliography

[1] Denise L. Wilson. *Theory of Signal Detection and its Application to Visual Target Acquisition: A Review of the Literature, AL-TR-1992-0083*. Armstrong Laboratory, Wright-Patterson Air Force Base, Ohio, May 1992.

[2] Feitshans, G., A. Rowe, J. Davis, M. Holland, and L. Berger. "Vigilant Spirit Control Station (VSCS) 'The Face of COUNTER'". *Proceedings of the AIAA Guidance, Navigation and Control Conference*. 2008.

[3] Green, D. "Psychoacoustics and Signal Detection Theory". *The Journal of the Acoustical Society of America*, 32:1189–1203, 1960.

[4] Green, D. and J. Swets. *Signal Detection Theory and Psychophysics*. John Wiley & Sons, New York, 1966.

[5] Gross, D., S. J. Rasmussen, P. Chandler, and G. Feitshans. "Cooperative Operations in UrbaN TERrain (COUNTER)". *Proceedings of SPIE*, volume 6249 of *Defense Transformation and Network-Centric Systems*. 2006.

[6] Holsapple, R., P. Chandler, J. Baker, A. Girard, and M. Pachter. "Autonomous Decision Making with Uncertainty for an Urban Intelligence, Surveillance and Reconnaissance (ISR) Scenario". *Proceedings of the AIAA Guidance, Navigation and Control Conference*. 2008.

[7] Pachter, M., P. Chandler, and S. Darbha. "Optimal MAV Operations in an Uncertain Environment". *International Journal of Robust and Nonlinear Control*, 00:1–6, 2006.

[8] Pachter, Meir, Philip Chandler, and Swaroop Darbha. *Optimal Control of an ATR Module Equipped MAV/Human Operator Team*. WSPC Proceedings, Jun 2006.

[9] Pachter, Meir, Phillip Chandler, and Swaroop Darbha. *Optimal Sequential Inspection*. Conference on Decision and Control, Dec 2006.

[10] Sorkin, R. and D. Woods. "Systems with Human Monitors: A Signal Detection Analysis". *Human-Computer Interaction*, 1:49–75, 1985.

[11] Swanson, Leah, Eric Jones, Brian Riordan, Sylvain Bruni, Nathan Schurr, Seamus Sullivan, and Jonathan Lansey. "Exploring human error in an RPA target detection task". *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 2012 56: 328*. 2012.

[12] Tanner, W., J. Swets, and D. Green. *Some General Properties of the Hearing Mechanism*. Technical Report 30, University of Michigan, Electronic Defense Group, Ann Arbor, MI, 1956.

[13] Verghese, P. “Visual Search and Attention: A Signal Detection Theory Approach”. *Neuron*, 31:523–535, 2001.

[14] Warm, J., W. Dember, and P. Hancock. “Vigilance and Workload in Automated Systems”. R. Parasuraman and M. Mouloua (editors), *Automation and Human Performance: Theory and Applications*. Erlbaum, Hillsdale, NJ, 1996.

Appendix A: Choosing Stimuli for Target Identification

The initial concept for the target identification stimuli was to have a generic human figure as the object and the discriminating feature is whether the figure is carrying a weapon (rifle, Rifle Propelled Grenade (RPG)) or an implement (shovel, broom, pipe, etc). This brought extensive modeling and simulation issues to the forefront that threatened to overwhelm a proof-of-concept solution approach. The object configuration decided upon was a van with a **Y** or **V** letter on the side.

Also, it is widely known that the strongest feature recognition factors are pixels/feature (resolution) and color difference (contrast). A number of different van colors were tried (such as red, blue) and it was seen that there were statistically significant differences in operator performance due to background contrast. It was decided to use the off-white color to avoid having to ensure the colors were uniformly distributed over the observation (state) space. Similar issues were found with different classes of vehicles, such as van and box truck. The box truck feature discrimination errors were lower than the errors for the van, although the feature size and colors were comparable. So, for this reason the box truck was not used.

What this means is, if one wants to expand the possible feature variations so that the results are more robust, then many more samples are required to cover the state space [Ground Sample Distance (GSD),Aspect Angle (AA),Dwell Time (DT),Inter-Event Time (IET)]. Even so, the resultant operator error model will be an average over the feature variations, and could be quite far from a specific color (contrast) sample. For these reasons, all of the objects are vans, the color is off-white, and the features are **Y** and **V**.

Appendix B: Video Complications

When the operator selects Target/NonTarget, the alert is reset (turned off). Unfortunately, this does not always happen. For reasons that were never completely isolated, the alert will sometimes stay on. This could confuse the operator, but the alert will reset if Target/NonTarget is pressed again. The test director monitors the alerts and will tell the operator to press again. After a little experience with this anomaly, the operator would reliably reset as needed. The operator felt this is a minor distraction and did not detract from his performance.

All of the videos consistently experienced a low frequency (about 1Hz) jitter or hesitation. The concern was that this anomaly would compromise the operator performance and the validity of the results. An investigation was undertaken to determine the source of the video jitter. There was no jitter at the graphic computers, only when the video was displayed in the Vigilant Spirit Control Station (VSCS) media players. The media players parameters were varied (at the expense of delay), and this did help, but the jitter could be reduced, not eliminated. At this point, it was estimated that a significant effort would be required to track down and fix the jitter. However, further experimentation with the operator found, that while the jitter was annoying, it caused little deterioration in the operator performance. The Tactical Situation Display (TSD) also had an effect on the video jitter. If the exploded view with the path is used, then this would aggravate the video jitter. The cause was never found, but the symptom was persistently observable. In summary, the TSD was not used by the operator for the feature classification task. For these reasons, the effort pressed on with the video jitter in place.

For the operator, the inspection task is quite different from the search task. The search task is higher workload, while the cued inspection is less intense. The alert tells the operator where and when to look. The alert is effective, as the operator never failed to note that it was on. This certainly impacts the Missed Detection rate, as well as the nature of MD. For the search task, a MD is truly a “Miss”. The operator was looking elsewhere and didn’t see it. A “Miss” for the inspection task is actually a misclassification. The operator saw the object, but misclassified the feature. For the scenarios in this report, the operator essentially did not have any misses—meaning that the operator always provided an input for every inspection task. This is of course important for the data collection, as more cases would have to be run to make up for the ones missed. But not missing is also critical for the SC. This is because a miss means no information was gained for that inspection task. With no information, the Unmanned Aerial Vehicle (UAV) will proceed to the next inspection task in the sequence. This will skew the results if one tries to compare one run with another. But does wash out with large numbers.

The plan tracking by the autopilot was uniformly excellent. The object was consistently in the middle of the camera footprint (video), and went completely through from top to bottom. This would have been difficult, if not impossible, to achieve with a fixed camera if there was wind and/or gusts. Also, all of the inspection tasks were reviewed to ensure that the feature was always visible. The operator did not have a “Can’t See” choice.

Appendix C: UAV Definition

The Vigilant Spirit Control Station (VSCS) has a number of Unmanned Aerial Vehicle (UAV)s that can be simulated. The one used here is similar to the Bat3. It is gas powered, has a wingspan of 9 feet, and a gimballed camera. For the purposes of this effort, the most critical specifications are the speed range [15 knots–50 knots], turn rate [2 g's], and the camera [fixed]. Speed and turn rate are needed to compute the turn radius, which is needed to determine the Dubins path. The sensor geometry is shown in Figure C.1. The camera is fixed to limit the degrees of freedom in sensor planning that would be introduced with a gimballed camera. As can be seen in Figure C.1, the camera has a fixed depression angle θ of 45 degrees. The vertical field of view (VFOV) is 22 degrees, and the horizontal FOV (HFOV) is also 22 degrees. This gives a reasonably rectangular (trapezoid) footprint. This footprint will vary in size as a function of altitude above ground level (AGL=h). The

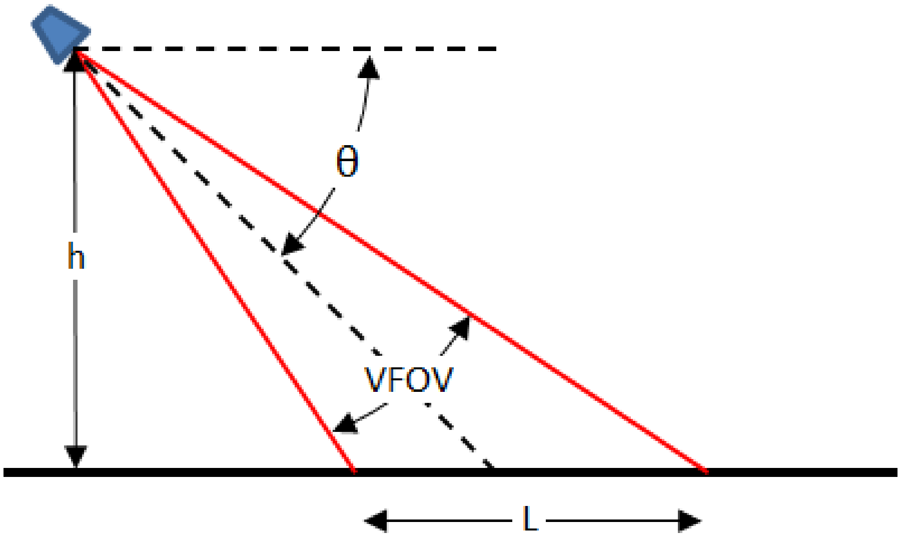


Figure C.1: Sensor Geometry and Footprint

Above Ground Level (AGL) and footprint gives Ground Sample Distance (GSD) and/or pixels-on-target. GSD is more generic and is used as one of the independent variables in the operator error (performance) model. The equation for GSD is:

$$GSD = FOV \cdot \frac{AGL}{\cos\theta}$$

The equation for the footprint length is:

$$L = \frac{h}{\tan(\theta - VFOV/2)} - \frac{h}{\tan(\theta + VFOV/2)}$$

The Dwell time is then the footprint length L divided by the UAV speed. Note that the Dwell time is not the operator response time.

There is no wind in the UAV simulation, so that the air speed = ground speed, and UAV heading = ground track heading (no sideslip). This greatly simplifies the path planning problem and the ability of the autopilot to track the plan. While having wind would be more realistic, it greatly increases the variability in the data gathering (model) and the SC evaluation. It is much more difficult to keep the Target in the footprint. This would necessitate many more samples to ensure that sufficient samples were available to cover the Dwell times in the operator model.

However, with a body axis fixed sensor, the footprint cannot remain on the plan while turning. This would require a gimballed sensor. Therefore, a standoff distance must be enforced, before the Target is reached, for the UAV to be on the planned path and wings level.

Appendix D: Operator Briefing

Operator Training Bullets

Interface layout

• Left monitor is the “God’s eye” view of the map. It shows all of your UAVs and their locations. Not really necessary to focus on the map for these trials because the other monitor will alert you to an approaching point of interest.

• Right monitor is the 4 camera views of the UAVs: MUSCIT 21 (top left), MUSCIT 22 (top right), MUSCIT 23 (bottom left), MUSCIT 24 (bottom right).

• “Target” and “Non-Target” buttons to the right of each camera view are the buttons that apply to that respective camera.

• The operator is to sit 32inches from the display screen (eyeball to screen), which is marked on the side table. The operator is not to squint or lean forward.

Task

• Action: Provide participant with the pictures that show the vehicles with the **Y** and **V** shapes

• Press the “Target” button when you see a vehicle with a **Y** shape.

• Press the “Non-Target” button when you see a vehicle with a **V** shape.

• One way to distinguish between Targets from Non-Targets is the stem at the bottom of the letter.

• If there is some uncertainty as to whether or not the vehicle is a target or non-target, simply make your best guess. (Unless requested otherwise.)

• All vehicles will be white and the letters will be black and upright.

• Area surrounding the buttons will cue you by highlighting “red” approximately 2-3 seconds before the UAV passes over the vehicle of interest.

- Could have multiple camera views highlight red simultaneously (or close to simultaneously). You are still responsible for classifying all vehicles so do your best to identify each vehicle as target or non-target.
- For each button push, the operator is to render a number from 1–5, signifying the operator's confidence in the Target/NonTarget decision. (1-no confidence, 2-low confidence, 3-moderate confidence, 4-high confidence, 5-high confidence.) This will be recorded by the test director.

Miscellaneous

- Let the test director know if you accidentally hit the wrong button and you "without a doubt" know that you hit the wrong button.
- The buttons, on occasion, will remain highlighted red or immediately return to a red highlight. Simply hit the button again. The test director will monitor.
- Target density is approximately 50%.
- The operator is to have 20-20 vision and not be color blind.
- Each run will be approximately 12 minutes long.

Training threshold

- Set at 70% average over all runs for Ptr and Pftr.

Appendix E: Experiment 1 Pilot Data: Operator S0

The instructions to the operator by the test supervisor are given in Appendix D. The operator was told to "do the best you can" and no performance feedback was provided.

E.0.1 Test0: Familiarization - baseline, 50% density.

Run	Scenario	FA1%	MD1%	FA2%	MD2%	PC%
1	1A	30.00	55.56	33.33	0.00	57.50

Just to reiterate, Operator S0 looks at each of 20 objects in a Run and determines whether the object is a Target or a Nontarget. The single familiarization Run has a Target density of 50%, which always gives exactly 10 Targets and 10 NonTargets. There are 4 Scenarios, numbered 1–4, where the objects have different locations and values. The A in 1A signifies that this is the 1st occurrance of Scenario 1 in a operator data taking session. The FA1 signifies the 1st look False Alarm rate and MD1 is the 1st look Missed Detection rate. In this case, of the 10 NonTargets, the operator said 3 were actually Targets. Very rarely the operator fails to provide an observation on an object, and that is counted as a "miss", which is the case here. The MD1 is then computed as 5/9.

The FA2 and MD2 operator observations are for the 2nd look. The number of revisit opportunities are set in the Stochastic Controller, and for this project, set at 3. This means of the 5 objects that are inspected per Unmanned Aerial Vehicle (UAV), only 3 can be revisited to provide additional information. For the 4 UAVs in a Run, that gives a total of 12 objects that can be revisited. But the ratio can be 6T/6NT, as it is here, or it can be 3T/9NT, or 4T/8NT, and so forth. For this Run, the operator did not misclassify any of the Targets on revisit, but did misclassify a third of the NonTargets on revisit. The

Percent Correct, or PC performance metric was not used until later, but is introduced here for continuity. The Percent Correct (PC) is simply the percentage of the objects that the operator classified correctly. This can be computed as $PC = ((100 - FA) + (100 - MD))/2$. Previous research has suggested that if the operator does not favor Targets or NonTargets, then the operator is not "biased" and will try to maximize PC. Unless stated otherwise, for all the Runs the operator is instructed to "do the best you can", which is interpreted to mean maximize PC.

The problem these series of tests was intended to address was very low False Alarm (FA) and Missed Detection (MD) rates, caused by the scenarios being too easy. The low performance in Test0 of PC=57.5%, or nearly random, could be ascribed to the operator never having seen the NonTarget feature **V** until the familiarization run was underway.

E.0.2 Test1/2: 50ft+, fast, 50% density.

Run	Scenario	FA1%	MD1%	FA2%	MD2%	PC%
1	1A	10	10	0	10	90
2	2A	30	30	75	0	70
3	3A	30	30	0	0	70
4	4A	10	20	0	0	85
5	1B	10	10	0	0	90
6	2B	10	10	0	0	90
7	3B	10	10	0	11	90
8	4B	30	20	0	12	75

This test constituted the baseline for the remaining tests. For the 8 runs, the average FA1=17% and the average MD1=17%. The average PC=83% with reasonably consistent results between runs, given that this is a stochastic process with significant variations. The operator performance at the revisit state is excellent except for run 2 where FA2=75%. This may be an outlier. The operator model called `LEAPModel13_plus50ft.xml` is used by the SC for the revisit state calculation, where the model is a modification of the LEAP II operator model called `LEAPModel13.xml`. There is an unknown amount of error between Operator S0 and `LEAPModel13_plus50ft.xml`. However, the revisit state appears to be a good choice. In summary, this condition appears to be a reasonably good point of departure.

E.0.3 Test3: 50ft+, fast, 50% density, biased.

Run	Scenario	FA1%	MD1%	FA2%	MD2%	PC%
1	1A	50	0	0	0	75
2	2A	50	10	0	0	70
3	3A	0	20	0	22	90
4	4A	0	20	0	0	90

Test3 is the same as Test1/2 except the operator was instructed to be more liberal. In others words, when the operator is in doubt, say Target. One would think that this instruction would reduce the MD1 rate and drive up the FA2 rate. And initially, this seems to be what happened in Scenario 1A and 2A. However, in Scenario 3A and 4A the results went exactly in the other direction. On average, FA1=25%, MD1=12%, and PC=81%. However, there are large fluctuations. If the trendlines continue, then the MD rate will be increasing and the FA rate will be decreasing. One explanation is that the operator is having trouble calibrating the bias, and the prediction is the operator performance will be erratic. With no feedback, the operator bias could change dynamically in longer runs. Therefore, it was decided not to pursue the "bias" line of investigation to increase the FA rate at the present time.

E.0.4 Test4: 25ft+, fast, 50% density.

Run	Scenario	FA1%	MD1%	FA2%	MD2%	PC%
1	1A	10	10	0	0	90
2	2A	0	0	0	0	100
3	3A	20	10	0	0	85
4	4A	0	0	0	0	100

Test4 was run to determine the sensitivity to the baseline altitude of 50ft+. This was changed to 25ft+, which is a net reduction in altitude of 25ft. The operator performance improved from the baseline of FA=MD=17% to FA=7%, MD=5%, and PC=94%. This was more than expected, so operator performance is quite sensitive to altitude. Increasing the altitude even further is done in Test8/9. Tests6/7 investigate changes in density.

E.0.5 Test5: familiarization - 50ft+, fast, 5% density.

Run	Scenario	FA1%	MD1%	FA2%	MD2%	PC%
1	1A	26	0	0	0	87

This is the first low density scenario that Operator S0 has seen, and for this reason it was decided to treat it separately as a familiarization run. The operator was not told that the scenario has 5% target density, or even that it is low density. It appears the operator proceeded as if the scenario was high density—this would explain the FA1=26% result. There is only one Target, so MD1 will be either 0% or 100%. The operator was fortunate in this case. The next test, Test6, should give a truer estimate of the operator error at low density.

Run	Scenario	FA1%	MD1%	FA2%	MD2%	PC%
1	1A	5	0	0	0	97.5
2	2A	10	0	0	0	95
3	3A	5	0	0	0	97.5
4	4A	10	0	0	0	95
5	1B	15	100	8	0	42.5
6	2B	5	100	0	0	47.5
7	3B	0	0	0	0	100
8	4B	0	0	0	0	100
9	1C	0	0	0	0	100
10	2C	5	100	0	0	47.5

E.0.6 Test6: 50ft+, fast, 5% density.

The average FA1=5.5%, MD1=33%, and PC=80.75%. However, there are only 10 Target samples in Test6. Many more tests would need to be run to be able to make a confident estimate of MD1. This is a fundamental problem with all low density scenarios—few target observations. While the distribution of Type I and Type II errors have changed, the PC=80.75% is quite close to the baseline PC=83%. After the familiarization run, the operator felt that there were 1,2, or 3 Targets in each run. In actuality, there were 1 or 2 Targets viewed, depending on whether the controller revisited the single target. The operator said that if the Target came early in the scenario, then felt confident in saying rest are not Targets. If the Target came late, then had a tendency to have 1 or 2 FAs.

E.0.7 Test7: 50ft+, fast, 25% density.

Test7 has 25% density with 15 NonTargets and 5 Targets per run. There is at least 1 Target per UAV and 1 UAV will have 2 Targets. There are not enough scenarios so that all the combinations of UAV and position are represented. The average FA1=2%, MD1=14%, and PC=92%.

To summarize: at 50% density, PC=83%; at 25% density, PC=92%; and at 5% density, PC=80%. It is not at all clear why PC would increase so significantly at 25% density.

Run	Scenario	FA1%	MD1%	FA2%	MD2%	PC%
1	1A	0	0	0	20	100
2	2A	6.7	20	0	25	86.5
3	3A	0	20	0	0	90
4	4A	0	20	0	0	90
5	1B	0	20	0	20	90
6	2B	0	40	0	20	80
7	3B	6.7	0	0	0	96.7
8	4B	0	0	0	0	100
9	1C	0	20	0	0	90
10	2C	6.7	0	0	0	96.7

However, the total number of errors appears to decrease with density. At 50% density there are 17.5 errors per 100 objects; at 25% density there are 5 errors per 100 objects; and at 5% density there are 7 errors per 100 objects. Clearly the operator's performance changes with density. Further investigations of density effects were curtailed in favor of investigating additional increases in altitude to increase FA rate.

E.0.8 Test8: 100ft+, fast, 25% density.

Run	Scenario	FA1%	MD1%	FA2%	MD2%	PC%
1	1A	6.7	40	0	0	77
2	2A	6.7	0	0	0	96
3	3A	6.7	0	0	0	96
4	4A	13	20	0	0	83
5	1B	6.7	0	0	20	96

Test8 raises the altitude by 100ft at 25% density. Test8 has average FA=8%, MD=12%, and PC=90%. There is an interaction between the increase in altitude and the decrease in density from the baseline. Here, both FA and MD went down, with FA going down much more than MD. The main difference from Test7 is that the altitude increase caused the FA rate to increase from 2% to 8%.

E.0.9 Test9: 100ft+, fast, 50% density.

Run	Scenario	FA1%	MD1%	FA2%	MD2%	PC%
1	1A	10	20	0	0	77
2	2A	10	10	0	11	96
3	3A	0	30	0	11	96
4	4A	10	30	0	0	83
5	1B	0	20	0	20	96

Test9 is the last perturbation in altitude and density to determine their effect on operator performance. At 100ft+ increment in altitude and 50% density, average FA=6%, MD=22%, and PC=86%. Compared to the baseline Test1, FA rate decreased by a factor of 4, the MD rate increased by 30%, and PC was about the same. The 5 runs of data is not a big enough sample to draw conclusions, but the tentative observation is that increasing the altitude as well as decreasing the density has the effect of significant decreases in FA rate. This, of course, does not achieve the desired effect of increasing FA rate, where a FA=20%–25% was the nominal design point for the Stochastic Controller.

E.0.10 Test Results Summary.

Date	Density	Δ Altitude	Speed	Bias	FA	MD	PC
11-12 Apr	50%	+50	+	”best”	17%	17%	83%
16 Apr	50%	+50	+	”liberal”	25%	12%	81%
17 Apr	50%	-25	+	”best”	7%	5%	94%
22-23 Apr	5%	+50	+	”best”	5%	30%	82%
8-9 May	25%	+50	+	”best”	2%	14%	92%
16 May	25%	+100	+	”best”	8%	12%	90%
20 May	50%	+100	+	”best”	6%	22%	86%

Appendix F: Interim Models for Operator S1

F.1 `LEAPmodel0808.xml` with 25 data runs

This section constitutes the first 25 runs of the total 120 runs to develop the final operator error model. The approach taken in these 120 data runs does not follow a more conventional approach as typified in LEAP II. The conventional approach would separate the estimation and control and perform them sequentially in different simulations. The model estimation would consist of the development of model scenarios that consisted of 16 discrete states that are composed of the high and low values of the 4 independent variables of Ground Sample Distance (GSD), Aspect Angle (AA), Dwell Time (DT), and Inter-Event Time (IET). A relatively complex bookkeeping scheme is needed to ensure there are 100 samples at each of the 16 states for both the Target and NonTarget. This is a more difficult task than it looks, given that IET in particular is a difficult variable to program independently. Once the data is available, a model can be fitted—of which there are many methods. Then, the model is inserted into the Stochastic Controller as part of a different evaluation simulation. This stochastic simulation has continuous independent variables that dynamically change the revisit state and the IET. The distributions of the model data and the evaluation data can be, and generally will be, quite different. This can result in significant modeling error and resultant drop in performance.

The approach pursued here is to simultaneously perform estimation and control in the exact same simulation. This dual control problem exploits the separation in operator error data and the evaluation data. The operator error model strictly uses the first look states, which are (nearly) independent from the second look states commanded by the controller. They are not completely independent because they are coupled by the IET. However, the operator model is only a weak function of IET. Therefore, changing the model should have little effect on the distribution of the model data in later runs. This has indeed been verified in practice in later sections.

The original plan was to update the operator model after 25, 50, 75, 100, 125, and 160 runs. But in practice the individual estimates stabilized quicker than expected—this resulted in updating the model at 25 runs, at 75 runs, and at 120 runs. The first 25 runs were taken on 2,5,6,7,8 Aug 2013. Prior to the beginning of these and all subsequent runs, the object vehicles were changed. Previously there were 2 object vehicles, a van and a truck. The truck and the corresponding feature is larger and therefore the operator has lower net errors in classifying features on the truck versus the van. Since there are only 4 scenarios, neither vehicle was seen by the operator at all the 16 states. The error statistics are for a mixture. It was decided to remove the truck and replace it with the van in all 4 of the scenarios to ensure consistency. This, however, increased the error rate above the target limit of FA<25%. It was decided to continue the tests at the higher error rate.

For the 25 runs, the overall FA=34%, MD=15%, and PC=76%. As previously discussed, this is above the FA target. The closed loop performance has Estimator FA=5%, MD=10%, and PC=92%. The closed loop performance is calculated from the operator error model, which for these runs, is `LEAPModel13_225to325ft224_455s.xml`. The error statistics for this model are quite similar to the error statistics for the present 25 data runs (which was not planned). Therefore, it could be extrapolated that the Estimator performance using an operator model derived from 120 runs will be similar to the Estimator performance here. And so it has proved.

The first 10 runs have FA=[20 60 10 40 40 40 40 40 20 40].

The first 10 runs have MD=[30 0 20 30 20 11 20 0 20 10].

The first 10 runs have PC=[75 70 85 65 70 75 70 80 80 75].

While there is significant variation run to run, averaging over a 10 run horizon quickly converges to the long term average. FA10=35%, MD10=16%, and PC10=75%, which is only 1% off the 25 run average. A minimum of 10 runs is used to report average statistics.

F.1.1 Operator Model `LEAPmodel0808.xml`*.*

The histograms below show the the distribution of the 4 independent variables that are used to make the operator error model. In Figure F.1 can be seen, from the data, that altitude varies from low 225ft Above Ground Level (AGL) to high 325ft AGL which is equivalent to low GSD=0.1909ft/pixel and high GSD=0.2757ft/pixel respectively for the fixed camera. In the Vigilant Spirit Control Station (VSCS) Cooperative Control Algorithms (CCA) simulation, the GSD is uniformly sampled between the high and low values. The simulation takes the low and high values from the operator error model itself. The actual achieved values, as shown in the example data file in Appendix I, are within a couple of feet of the planned. The actual values are what is used in the operator model data. The distribution is quite reasonable for 500 random samples.

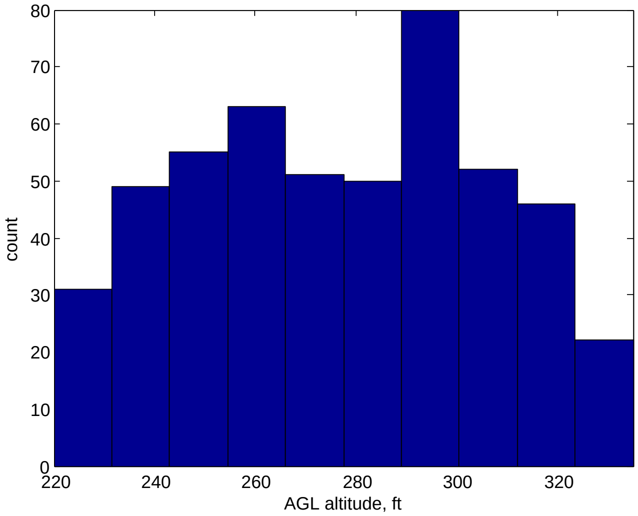


Figure F.1: Altitude Distribution

Approved for public release; distribution unlimited.

In Figure F.2 can be seen the Aspect Angle data which varies from low 45deg to high 90deg. The simulation also takes the high and low values from the operator error model. The uniform random distribution is closely approximated by the data.

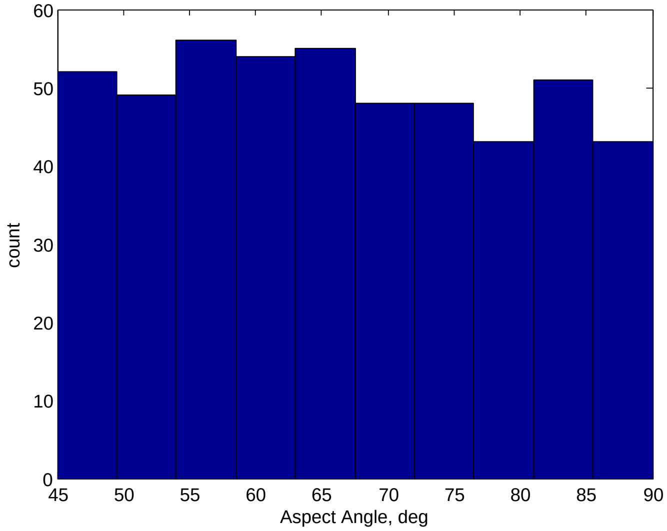


Figure F.2: Aspect Angle Distribution

In Figure F.3 is the data for the Dwell Time, or the time duration of the object in the viewing frame. This varies from a low 2.24sec to high 4.55sec. The data closely approximates a uniform random distribution.

Finally, Figure F.4 shows the Inter Event Time data distribution, where IET is the time between subsequent viewings of objects. The IET is not controlled in the simulation and the low and high values are not used in the CCA planner. The IET is the result of the planner and the controller. It is not realistic to try and control the IET. As can be seen, it is not even approximately uniform. The approach is to use the median of the data (15sec) to divide between low and high.

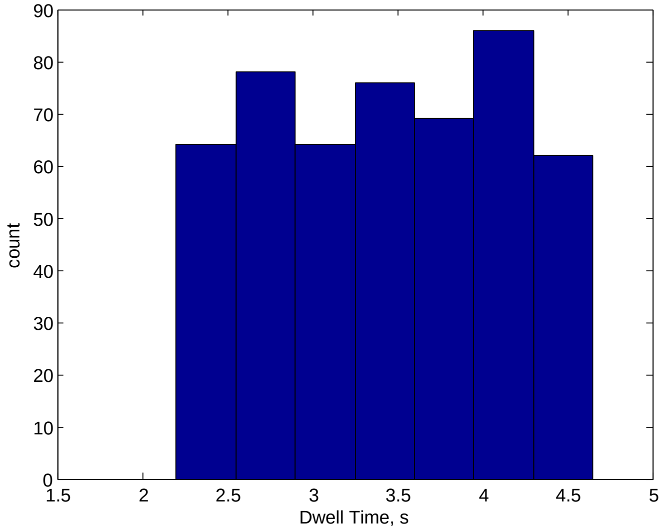


Figure F.3: Dwell Time Distribution

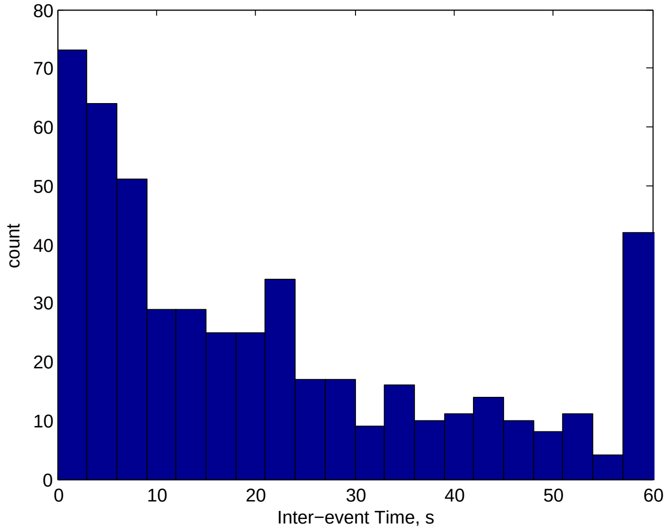


Figure F.4: IET Distribution

To create the operator error model, the data is divided into bins. The midpoint of the independent variables is GSD=0.233, AA=67.5, Dwell=3.395, and IET=15. The resulting

16 bin combinations of the data are shown in Table F.1 in columns 2–5. Since 25 runs, at 50% density, has 250 Targets and 250 NonTargets, then column 6 and column 9 add up to 250 (unless there is a "miss"). What is important is the distribution of the samples over the 16 states, which if perfectly distributed, is 16 samples/state. The NonTarget distribution has a max of 22 samples for state 9, a min of 10 samples at state 5, a mean of 15.6 samples, and Standard Deviation (std) of 4.1 samples. The Target distribution has a max of 26 samples at state 15, a min of 7 samples at state 1, a mean of 15.6 samples, and std of 5.4 samples. The NonTarget distribution is excellent, and the Target distribution, while not excellent, is still quite good. There are no 0 sample bins and no bins that are highly over represented. There are, however, 2 states, 6 and 8, that have no hits and consequently has FA identically equal to 0. These "perfect" states could bias the stochastic controller as well as cause an error in the mutual information calculation. For these reasons, the calculation is made where FA=$.5/n$ where n is the number of samples. A similar problem occurs for MD at state 7.

Table F.1: Operator Model Data F0808

State	GSD	Angle	Dwell	IET	Samples	Hits	FA	Samples	Hits	MD
1	0.1909	45	2.24	0	21	6	0.2857	7	6	0.1428
2	0.1909	45	2.24	30	18	4	0.2222	16	15	0.0625
3	0.1909	45	4.55	0	11	3	0.2727	11	9	0.1818
4	0.1909	45	4.55	30	13	2	0.1538	24	21	0.125
5	0.1909	90	2.24	0	10	2	0.2	14	11	0.2142
6	0.1909	90	2.24	30	14	0	0.0357	10	9	0.1
7	0.1909	90	4.55	0	12	2	0.1666	14	14	0.0357
8	0.1909	90	4.55	30	14	0	0.0357	9	8	0.1111
9	0.2757	45	2.24	0	22	9	0.4090	20	16	0.2
10	0.2757	45	2.24	30	16	13	0.8125	16	15	0.0625
11	0.2757	45	4.55	0	20	9	0.45	21	15	0.2857
12	0.2757	45	4.55	30	16	9	0.5625	14	13	0.0714
13	0.2757	90	2.24	0	13	6	0.4615	19	15	0.2105
14	0.2757	90	2.24	30	10	4	0.4	12	11	0.0833
15	0.2757	90	4.55	0	21	8	0.3809	26	21	0.1923
16	0.2757	90	4.55	30	19	8	0.4210	16	14	0.125

For the operator error model Table F.1, the overall FA=33%, MD=14%, and PC=76.5%. There is no more than a 1% difference with the overall data results that were discussed earlier. This means that the operator error model is an overall good fit to the data. The individual state values will improve over time as more and more data is added to the model. This will be seen in later sections. It is fairly clear that the 8 low altitude states have a much lower error than the 8 high altitude states. This agrees with intuition. The "best" states are 7,8 and the "worst" states are 9,10—which also agrees with intuition. These results will be consistently true over the next 2 models. The low altitude states and the high altitude states appear to be fairly consistent for MD. There does not appear to be a standout "bad" state for MD. This is not the case for FA, where all the high altitude states are ≥ 40%. This means that the operator, for a NonTarget at high altitude, is essentially guessing. The exception at state 10 has FA=81%. This means the operator is a contra-indicator—if the operator says it's a Target, there's a high probability it isn't. As will be seen, as more data becomes available for later models, these probabilities will reduce. Also, the revisit state computed by the Stochastic Controller remains consistently the same for all 3 of the operator error models.

F.2 `LEAPmodel0828.xml` with total 75 data runs

Analysis of Model Change with 10 runs

The first action taken here is to verify that the 1st look data used for the operator model is reasonably independent from the controller. This would mean that the operator model can justifiably be incrementally updated over the data runs, yet all the 1st look data can be used for the new operator error model. The 10 runs before the 0808 model update is compared to the 10 runs just after the 0808 model update. The prior 10 runs from 201307,08 have operator FA=31%, MD=16%, and PC=76.5%. The 10 runs from 20130813,14 just after the 0808 model update have operator FA=35%, MD=25%, and PC=70%. This is enough of a difference to warrant investigation. If you look at the PC for these 10 runs, you get

PC=[70 70 75 65 75 80 65 40 70 90]. Run 8, or run labeled 20131431, looks like it may be an outlier with a PC=40%. If the 40% is replaced with the prior mean of 75%, then the new mean of the 20130813,14 data set is PC=74%. This is quite reasonable compared to the 201307,08 data set PC=76.5%. Also, the std of the new augmented data set is 7.4%, which means the 8th data run with PC=40% is almost 5 std out. A metric of 3 std is used to define an outlier. Further evidence of an outlier is that run 20131431 has 2nd look FA=57% and MD=20%. This is by far the largest 2nd look error rate in all of the 120 runs. The surmise is that there was some as yet unknown error in setup or simulation. This event has not been repeated. When 20131431 has been removed from the data set, then operator FA=33%, MD=23%, and PC=71.7%.

While the FA rate before and after model change is about the same, the MD has increased from 16% to 23%. This is not an insignificant change. Further investigation revealed some rounding error in GSD that resulted in approximately a 4ft error in the GSD limits. This was not thought to be significant. There may indeed be more coupling between the estimator and the controller, and/or the data may not be as representative as thought. Regardless, the approach appears to be working reasonably well and the decision was made to continue taking data with model 0808.

Analysis of Model Change with 25 runs

If you look at the 25 runs before and after the model change, you get:

Before; operator FA=34%, operator MD=15.5%

After; operator FA=35.8%, operator MD=20%

Note, the after statistics actually use 24 runs with the outlier taken out. This is similar to the previous 10 run comparison, where there seems to be a persistent increase in the MD rate. But essentially no change in the FA rate. Some of this may still be from stochastic variation, but it looks less likely as the number of samples increase. There may be some slight change in the operator bias that is somehow correlated with the model change. These effects will be washed out as the number of samples is increased. Interestingly, there is no change at all in the estimator performance:

Before; estimator FA=5.2%, estimator MD=10.4%

After; estimator FA=5.8%, estimator MD=10.5%

This indicates that the Estimator is robust to limited variations in the operator error model.

Analysis of Model Change with 50 runs

This analysis is for 50 runs from 20130813 to 20130828. The comparison is now:

Before; operator FA=34.0%, operator MD=15.5%, operator PC=75%

After; operator FA=37.6%, operator MD=18.0%, operator PC=72%

While only 25 runs are available before the model change, the conjecture can be made that if 50 runs had been run before the change and the results compared to the 50 runs after the change, then there would be little change. The conclusion can be made that the model change has made little change in the operator data, and the concurrent approach to the estimator and controller is valid.

F.2.1 `LEAPmodel0828.xml`.

There were 75 total runs from 20130802 to 20130828 available to create the 0828 operator error model.

For the operator error model Table F.2, the overall FA=36%, MD=17%, and PC=73.5%. There are an average of 46 samples per state. There is still more variance between the targets states than the nontarget states. In general, there is no more than about 10% of full scale difference for FA and MD between 0808 and 0828. This indicates that a sufficient number of samples is being approached for a reasonable estimate of the individual states. At this point it was decided that 120 runs would probably be adequate for the final operator error model, which gives, on average, 75 samples per state.

Table F.2: Operator Model Data F0828

State	GSD	Angle	Dwell	IET	Samples	Hits	FA	Samples	Hits	MD
1	0.1909	45	2.24	0	52	21	0.4038	29	25	0.1378
2	0.1909	45	2.24	30	46	19	0.4130	43	40	0.0698
3	0.1909	45	4.55	0	27	6	0.2222	39	35	0.1026
4	0.1909	45	4.55	30	49	10	0.2041	52	46	0.1154
5	0.1909	90	2.24	0	30	3	0.1000	40	35	0.1250
6	0.1909	90	2.24	30	40	7	0.1750	32	30	0.0625
7	0.1909	90	4.55	0	41	3	0.0732	43	40	0.0698
8	0.1909	90	4.55	30	40	0	0.0125	40	39	0.0250
9	0.2757	45	2.24	0	51	27	0.5294	58	40	0.3103
10	0.2757	45	2.24	30	56	41	0.7321	38	32	0.1579
11	0.2757	45	4.55	0	54	21	0.3889	64	46	0.2813
12	0.2757	45	4.55	30	47	21	0.4468	39	32	0.1795
13	0.2757	90	2.24	0	47	23	0.4894	62	46	0.2581
14	0.2757	90	2.24	30	50	24	0.4800	44	34	0.2273
15	0.2757	90	4.55	0	52	22	0.4231	69	55	0.2029
16	0.2757	90	4.55	30	58	21	0.3621	45	38	0.1556

Appendix G: Operator Model matlab file

```
% ----------------------------------------------------------------------------
% BEGIN operator model m file

% This program creates the operator error model from the specified log files.
% Source file: analyzeStates5.m,  dated 11/5/2013, by pc

% The continuous state [GSD,AA,Dwell,IET] data from the logFiles
% is binned into 16 discrete states.  The FA and MD are tabulated
% for each of the states to yield Pfa and Pmd.

% Procedure
% Read logFiles
% Create 16 state model - ModelDataPR
% Create 256 state model - OpModelDataP256
% Create 256 state xml model - OpModelDataP256xxxx.xml

clear all
close all

% This represents the final configuration
% after the final change in the contrast.

% Runs 1-25 use LEAPModel13_225to325ft224_455s.xml
```

```
logFiles{1} = ’C:\...\S Final\20130802_1_1.csv’;
logFiles{2} = ’C:\...\S Final\20130802_2_1.csv’;
logFiles{3} = ’C:\...\S Final\20130802_3_1.csv’;
logFiles{4} = ’C:\...\S Final\20130802_4_1.csv’;
logFiles{5} = ’C:\...\S Final\20130802_1_2.csv’;

logFiles{6} = ’C:\...\S Final\20130805_1_1.csv’;
logFiles{7} = ’C:\...\S Final\20130805_2_1.csv’;
logFiles{8} = ’C:\...\S Final\20130805_3_1.csv’;
logFiles{9} = ’C:\...\S Final\20130805_4_1.csv’;
logFiles{10} = ’C:\...\S Final\20130805_2_2.csv’;

logFiles{11} = ’C:\...\S Final\20130806_1_1.csv’;
logFiles{12} = ’C:\...\S Final\20130806_2_1.csv’;
logFiles{13} = ’C:\...\S Final\20130806_3_1.csv’;
logFiles{14} = ’C:\...\S Final\20130806_4_1.csv’;
logFiles{15} = ’C:\...\S Final\20130806_3_2.csv’;

logFiles{16} = ’C:\...\S Final\20130807_1_1.csv’;
logFiles{17} = ’C:\...\S Final\20130807_2_1.csv’;
logFiles{18} = ’C:\...\S Final\20130807_3_1.csv’;
logFiles{19} = ’C:\...\S Final\20130807_4_1.csv’;
logFiles{20} = ’C:\...\S Final\20130807_4_2.csv’;

logFiles{21} = ’C:\...\S Final\20130808_1_1.csv’;
logFiles{22} = ’C:\...\S Final\20130808_2_1.csv’;
```

```
logFiles{23} = ’C:\...\S Final\20130808_3_1.csv’;

logFiles{24} = ’C:\...\S Final\20130808_4_1.csv’;

logFiles{25} = ’C:\...\S Final\20130808_1_2.csv’;


% From run 26-75 LEAPmodel88.xml is used.  Based on files 1-25.


logFiles{26} = ’C:\...\S Final\20130813_1_1.csv’;

logFiles{27} = ’C:\...\S Final\20130813_2_1.csv’;

logFiles{28} = ’C:\...\S Final\20130813_3_1.csv’;

logFiles{29} = ’C:\...\S Final\20130813_4_1.csv’;

logFiles{30} = ’C:\...\S Final\20130813_2_2.csv’;


logFiles{31} = ’C:\...\S Final\20130814_1_1.csv’;

logFiles{32} = ’C:\...\S Final\20130814_2_1.csv’;

logFiles{33} = ’C:\...\S Final\20130814_3_1.csv’;

logFiles{34} = ’C:\...\S Final\20130814_4_1.csv’;

logFiles{35} = ’C:\...\S Final\20130814_3_2.csv’;


logFiles{36} = ’C:\...\S Final\20130815_1_1.csv’;

logFiles{37} = ’C:\...\S Final\20130815_2_1.csv’;

logFiles{38} = ’C:\...\S Final\20130815_3_1.csv’;

logFiles{39} = ’C:\...\S Final\20130815_4_1.csv’;

logFiles{40} = ’C:\...\S Final\20130815_4_2.csv’;


logFiles{41} = ’C:\...\S Final\20130816_1_1.csv’;

logFiles{42} = ’C:\...\S Final\20130816_2_1.csv’;
```

```
logFiles{43} = ’C:\...\S Final\20130816_3_1.csv’;
logFiles{44} = ’C:\...\S Final\20130816_4_1.csv’;
logFiles{45} = ’C:\...\S Final\20130816_1_2.csv’;

logFiles{46} = ’C:\...\S Final\20130820_1_1.csv’;
logFiles{47} = ’C:\...\S Final\20130820_2_1.csv’;
logFiles{48} = ’C:\...\S Final\20130820_3_1.csv’;
logFiles{49} = ’C:\...\S Final\20130820_4_1.csv’;
logFiles{50} = ’C:\...\S Final\20130820_2_2.csv’;

logFiles{51} = ’C:\...\S Final\20130821_1_1.csv’;
logFiles{52} = ’C:\...\S Final\20130821_2_1.csv’;
logFiles{53} = ’C:\...\S Final\20130821_3_1.csv’;
logFiles{54} = ’C:\...\S Final\20130821_4_1.csv’;
logFiles{55} = ’C:\...\S Final\20130821_3_2.csv’;

logFiles{56} = ’C:\...\S Final\20130822_1_1.csv’;
logFiles{57} = ’C:\...\S Final\20130822_2_1.csv’;
logFiles{58} = ’C:\...\S Final\20130822_3_1.csv’;
logFiles{59} = ’C:\...\S Final\20130822_4_1.csv’;
logFiles{60} = ’C:\...\S Final\20130822_4_2.csv’;

logFiles{61} = ’C:\...\S Final\20130826_1_1.csv’;
logFiles{62} = ’C:\...\S Final\20130826_2_1.csv’;
logFiles{63} = ’C:\...\S Final\20130826_3_1.csv’;
logFiles{64} = ’C:\...\S Final\20130826_4_1.csv’;
```

```
logFiles{65} = ’C:\...\S Final\20130826_1_2.csv’;


logFiles{66} = ’C:\...\S Final\20130827_1_1.csv’;

logFiles{67} = ’C:\...\S Final\20130827_2_1.csv’;

logFiles{68} = ’C:\...\S Final\20130827_3_1.csv’;

logFiles{69} = ’C:\...\S Final\20130827_4_1.csv’;

logFiles{70} = ’C:\...\S Final\20130827_2_2.csv’;


logFiles{71} = ’C:\...\S Final\20130828_1_1.csv’;

logFiles{72} = ’C:\...\S Final\20130828_2_1.csv’;

logFiles{73} = ’C:\...\S Final\20130828_3_1.csv’;

logFiles{74} = ’C:\...\S Final\20130828_4_1.csv’;

logFiles{75} = ’C:\...\S Final\20130828_3_2.csv’;


% Used on runs 76-120, the OpModeldata250F0828 based on files 1-75.


logFiles{76} = ’C:\...\S Final\20130830_1_1.csv’;

logFiles{77} = ’C:\...\S Final\20130830_2_1.csv’;

logFiles{78} = ’C:\...\S Final\20130830_3_1.csv’;

logFiles{79} = ’C:\...\S Final\20130830_4_1.csv’;

logFiles{80} = ’C:\...\S Final\20130830_4_2.csv’;


logFiles{81} = ’C:\...\S Final\20130904_1_1.csv’;

logFiles{82} = ’C:\...\S Final\20130904_2_1.csv’;

logFiles{83} = ’C:\...\S Final\20130904_3_1.csv’;

logFiles{84} = ’C:\...\S Final\20130904_4_1.csv’;
```

```
logFiles{85} = ’C:\...\S Final\20130904_1_2.csv’;


logFiles{86} = ’C:\...\S Final\20130905_1_1.csv’;
logFiles{87} = ’C:\...\S Final\20130905_2_1.csv’;
logFiles{88} = ’C:\...\S Final\20130905_3_1.csv’;
logFiles{89} = ’C:\...\S Final\20130905_4_1.csv’;
logFiles{90} = ’C:\...\S Final\20130905_2_2.csv’;


logFiles{91} = ’C:\...\S Final\20130906_3_1.csv’;
logFiles{92} = ’C:\...\S Final\20130909_1_1.csv’;
logFiles{93} = ’C:\...\S Final\20130909_2_1.csv’;
logFiles{94} = ’C:\...\S Final\20130909_3_2.csv’;
logFiles{95} = ’C:\...\S Final\20130909_4_1.csv’;


logFiles{96} = ’C:\...\S Final\20130911_1_1.csv’;
logFiles{97} = ’C:\...\S Final\20130911_2_1.csv’;
logFiles{98} = ’C:\...\S Final\20130911_3_1.csv’;
logFiles{99} = ’C:\...\S Final\20130911_4_1.csv’;
logFiles{100}= ’C:\...\S Final\20130911_4_2.csv’;


logFiles{101}= ’C:\...\S Final\20130913_1_1.csv’;
logFiles{102}= ’C:\...\S Final\20130913_2_1.csv’;
logFiles{103}= ’C:\...\S Final\20130913_3_1.csv’;
logFiles{104}= ’C:\...\S Final\20130913_4_1.csv’;
logFiles{105}= ’C:\...\S Final\20130913_1_2.csv’;
```

```
logFiles{106}= ’C:\...\S Final\20130916_1_1.csv’;

logFiles{107}= ’C:\...\S Final\20130916_2_1.csv’;

logFiles{108}= ’C:\...\S Final\20130916_3_1.csv’;

logFiles{109}= ’C:\...\S Final\20130916_4_1.csv’;

logFiles{110}= ’C:\...\S Final\20130916_2_2.csv’;


logFiles{111}= ’C:\...\S Final\20130919_1_1.csv’;

logFiles{112}= ’C:\...\S Final\20130919_2_1.csv’;

logFiles{113}= ’C:\...\S Final\20130919_3_1.csv’;

logFiles{114}= ’C:\...\S Final\20130919_4_1.csv’;

logFiles{115}= ’C:\...\S Final\20130919_3_2.csv’;


logFiles{116}= ’C:\...\S Final\20130920_1_1.csv’;

logFiles{117}= ’C:\...\S Final\20130920_2_1.csv’;

logFiles{118}= ’C:\...\S Final\20130920_3_1.csv’;

logFiles{119}= ’C:\...\S Final\20130920_4_1.csv’;

logFiles{120}= ’C:\...\S Final\20130920_4_2.csv’;


% OpModelData256F0920.xml is based on all previous 120 files.



i_type = 1; %entry  (1 = first, 2 = revisit, 3 = alert set, 4 = in footprint)

i_time = 2;

i_vehID = 3;

i_vehHdg_deg = 4;

i_vehAltMSL_ft = 5;
```

```
i_POIID = 6;

i_POILat_deg = 7;

i_POILong_deg = 8;

i_POIAltMSL_ft = 9;

i_POIHdg_deg = 10;

i_AAdes_deg = 11;

i_GSDdes_ft = 12;

i_AAact_deg = 13;

i_GSDact_ft = 14;

i_DTact_s = 15;

i_IETact_s = 16;

i_PTR = 17; % Correct detection rate

i_PFTR = 18; % False alarm rate

i_PT1 = 19; % Prob of being a traget given report 1

i_PT2 = 20; % Prob of being a target given report 2

i_bTarget = 21; % Target truth data (only shows up on reports 1 and 2)

i_APD = 22; % Apriori density

i_revNum = 23; % Revisit number

i_revLeft = 24; % Revisits left

i_res = 25; % Operator response (1 = target, 0 = no target)

i_revDec = 26; % Revisit decision (0 = no revisit, >0 revisit state)

i_revAAdes_deg = 27; % commanded revisit Aspect angle

i_revGSDdes_ft = 28; % commanded revisit GSD

i_revAltdes_MSL_ft = 29; % Commanded revisit altitude

i_reviAA = 30;

i_reviGSD = 31;
```

```
i_reviDT = 32;
i_revIET = 33;
%i_cmdDT = 34; % commanded dwell time
%i_cmdSpeed = 35; % commanded speed
%i_actSpeed = 36; % actual speed
%i_confidence = 37; % operator confidence level 1-5

% Compile all runs
data = [];
i_logs = 1:120;
%i_logs = [26:32,34:50,51];
%i_logs = [26:32,34:60];
i_logs = [1:32,34:120];
%i_logs = [1:32,34:75];
for i = i_logs
%    newData  = readLog(logFiles{i});
    newData  = xlsread(logFiles{i},'a2:ak97');
    % using xlsread to get data from log file -- k, 7/31/2013
    % WARNING!! data in xcel sheet goes from column A to AK (37 columns)
    % and rows 2 to 97 (96 rows)
    data = [data; newData];
end

% ---------------------------------------------------------------------
% BEGIN extraction
first_looks = data(find(data(:,i_type) == 1),:);
```

```
first_FA = first_looks(find(first_looks(:,i_bTarget)==0...
           & first_looks(:,i_res)==1),:);
first_MD = first_looks(find(first_looks(:,i_bTarget)==1...
           & first_looks(:,i_res)==0),:);

revisits = data(find(data(:,i_type) == 2),:);
revisit_FA = first_looks(find(revisits(:,i_bTarget)==0...
             & revisits(:,i_res)==1),:);
revisit_MD = first_looks(find(revisits(:,i_bTarget)==1...
             & revisits(:,i_res)==0),:);
% END extraction
% ----------------------------------------------------------------

[hAlt hDwell hAA hIET] = plotStates(first_looks);



% -------------------------------------------------------------------------
% BEGIN state sorting

% Note: the target length and nontarget length need to be even?

% Separate the data into Target and Nontarget.
target_data = first_looks(find(first_looks(:,i_bTarget) == 1),:);
target_length = length(target_data);
target_length2 = round(target_length/2);
%sum(target_data(:,25));
```

```
nontarget_data = first_looks(find(first_looks(:,i_bTarget) == 0),:);
nontarget_length = length(nontarget_data);
nontarget_length2 = round(nontarget_length/2);
%sum(nontarget_data(:,25));

% Extracts states
states_target = [target_data(:,14),target_data(:,13),...
                 target_data(:,15:16),target_data(:,25)];

states_nontarget = [nontarget_data(:,14),nontarget_data(:,13),...
                    nontarget_data(:,15:16),nontarget_data(:,25)];

% Sorts by GSD
states_target_sort = sortrows(states_target);

states_nontarget_sort = sortrows(states_nontarget);
% END state sorting
% -------------------------------------------------------------------------



% ------------------------------------------------------------------------
% BEGIN grouping and sort for target data

% This section groups the target data using mean for g,a,d and median for i.
%  Note: should have used mean, but it works out to be about the same.
%gMid = (min(states_target_sort(:,1)) + max(states_target_sort(:,1)))/2;
```

```
gMid = 0.2333;

%aMid = (min(states_target_sort(:,2)) + max(states_target_sort(:,2)))/2;

aMid = 67.5;

%dMid = (min(states_target_sort(:,3)) + max(states_target_sort(:,3)))/2;

dMid = 3.395;

%iMid = median(states_target_sort(:,4));

iMid = 17.0;

%iMid = (min(states_target_sort(:,4)) + max(states_target_sort(:,)))/2;

%iMid = 10.0;

%dMid = 3.0;


% This line initializes the labels for each target row by bin number

% (binary 0-16) using column 6-9.

states_target_sort = [states_target_sort,9*ones(size(states_target_sort,1),4)];


% This section labels each target row by bin number.

for i=1:size(states_target_sort,1)

    if states_target_sort(i,1) < gMid

        states_target_sort(i,6) = 0;

    else states_target_sort(i,6) = 1;

    end

    if states_target_sort(i,2) < aMid

        states_target_sort(i,7) = 0;

    else states_target_sort(i,7) = 1;

    end

    if states_target_sort(i,3) < dMid
```

```
        states_target_sort(i,8) = 0;
    else states_target_sort(i,8) = 1;
    end
    if states_target_sort(i,4) < iMid
        states_target_sort(i,9) = 0;
    else states_target_sort(i,9) = 1;
    end
end

% This groups the target data by bin (binary) number from 0000 to 1111
r=find(states_target_sort(:,6)==0 & states_target_sort(:,7)==0...
  & states_target_sort(:,8)==0 & states_target_sort(:,9)==0);
StatesTargetMid = states_target_sort(r,:);
TargetStateSamples = length(r);
DetectionHits = sum(states_target_sort(r,5));
r=find(states_target_sort(:,6)==0 & states_target_sort(:,7)==0...
  & states_target_sort(:,8)==0 & states_target_sort(:,9)==1);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==0 & states_target_sort(:,7)==0...
  & states_target_sort(:,8)==1 & states_target_sort(:,9)==0);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==0 & states_target_sort(:,7)==0...
  & states_target_sort(:,8)==1 & states_target_sort(:,9)==1);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==0 & states_target_sort(:,7)==1...
  & states_target_sort(:,8)==0 & states_target_sort(:,9)==0);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==0 & states_target_sort(:,7)==1...
  & states_target_sort(:,8)==0 & states_target_sort(:,9)==1);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==0 & states_target_sort(:,7)==1...
  & states_target_sort(:,8)==1 & states_target_sort(:,9)==0);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==0 & states_target_sort(:,7)==1...
  & states_target_sort(:,8)==1 & states_target_sort(:,9)==1);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==1 & states_target_sort(:,7)==0...
  & states_target_sort(:,8)==0 & states_target_sort(:,9)==0);
```

```
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==1 & states_target_sort(:,7)==0...
 & states_target_sort(:,8)==0 & states_target_sort(:,9)==1);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==1 & states_target_sort(:,7)==0...
  & states_target_sort(:,8)==1 & states_target_sort(:,9)==0);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==1 & states_target_sort(:,7)==0...
  & states_target_sort(:,8)==1 & states_target_sort(:,9)==1);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==1 & states_target_sort(:,7)==1...
  & states_target_sort(:,8)==0 & states_target_sort(:,9)==0);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==1 & states_target_sort(:,7)==1...
  & states_target_sort(:,8)==0 & states_target_sort(:,9)==1);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
```

```
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==1 & states_target_sort(:,7)==1...
  & states_target_sort(:,8)==1 & states_target_sort(:,9)==0);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)];
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))];
r=find(states_target_sort(:,6)==1 & states_target_sort(:,7)==1...
  & states_target_sort(:,8)==1 & states_target_sort(:,9)==1);
StatesTargetMid = [StatesTargetMid;states_target_sort(r,:)];
TargetStateSamples = [TargetStateSamples,length(r)]
DetectionHits = [DetectionHits;sum(states_target_sort(r,5))]

% END grouping and sort for target data
% -------------------------------------------------------------------------

% Repeat for nontarget data

% -------------------------------------------------------------------------
% BEGIN grouping nontarget data
% This section groups the nontarget data using mean for g,a,d and median for i.
%  Note: should have used mean, but it works out to be about the same.
%gMid = (min(states_nontarget_sort(:,1)) + max(states_nontarget_sort(:,1)))/2;
gMid = 0.2333;
%aMid = (min(states_nontarget_sort(:,2)) + max(states_nontarget_sort(:,2)))/2;
aMid = 67.5;
```

```
%dMid = (min(states_nontarget_sort(:,3)) + max(states_nontarget_sort(:,3)))/2;

dMid = 3.395;

%iMid = median(states_target_sort(:,4));

iMid = 17.0;

%iMid = (min(states_nontarget_sort(:,4)) + max(states_nontarget_sort(:,)))/2;

%iMid = 8.0;

%dMid = 3.1581;

%aMid = 63.5432;


% This line initializes the labels for each nontarget row by bin number

% (binary 0-16) using column 6-9.

states_nontarget_sort = [states_nontarget_sort,...

                         9*ones(size(states_nontarget_sort,1),4)];


% This section labels each nontarget row by bin number.

for i=1:size(states_nontarget_sort,1)

    if states_nontarget_sort(i,1) < gMid

        states_nontarget_sort(i,6) = 0;

    else states_nontarget_sort(i,6) = 1;

    end

    if states_nontarget_sort(i,2) < aMid

        states_nontarget_sort(i,7) = 0;

    else states_nontarget_sort(i,7) = 1;

    end

    if states_nontarget_sort(i,3) < dMid

        states_nontarget_sort(i,8) = 0;
    else states_nontarget_sort(i,8) = 1;
    end
    if states_nontarget_sort(i,4) < iMid
        states_nontarget_sort(i,9) = 0;
    else states_nontarget_sort(i,9) = 1;
    end
end

% This groups the nontarget data by bin (binary) number from 0000 to 1111
r=find(states_nontarget_sort(:,6)==0 & states_nontarget_sort(:,7)==0...
  & states_nontarget_sort(:,8)==0 & states_nontarget_sort(:,9)==0);
StatesNonTargetMid = states_nontarget_sort(r,:);
NonTargetStateSamples = length(r);
FalseDetections = sum(states_nontarget_sort(r,5));
r=find(states_nontarget_sort(:,6)==0 & states_nontarget_sort(:,7)==0...
  & states_nontarget_sort(:,8)==0 & states_nontarget_sort(:,9)==1);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==0 & states_nontarget_sort(:,7)==0...
  & states_nontarget_sort(:,8)==1 & states_nontarget_sort(:,9)==0);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==0 & states_nontarget_sort(:,7)==0...
  & states_nontarget_sort(:,8)==1 & states_nontarget_sort(:,9)==1);
```

```
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==0 & states_nontarget_sort(:,7)==1...
  & states_nontarget_sort(:,8)==0 & states_nontarget_sort(:,9)==0);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==0 & states_nontarget_sort(:,7)==1...
  & states_nontarget_sort(:,8)==0 & states_nontarget_sort(:,9)==1);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==0 & states_nontarget_sort(:,7)==1...
  & states_nontarget_sort(:,8)==1 & states_nontarget_sort(:,9)==0);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==0 & states_nontarget_sort(:,7)==1...
  & states_nontarget_sort(:,8)==1 & states_nontarget_sort(:,9)==1);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==1 & states_nontarget_sort(:,7)==0...
  & states_nontarget_sort(:,8)==0 & states_nontarget_sort(:,9)==0);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
```

```
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==1 & states_nontarget_sort(:,7)==0...
  & states_nontarget_sort(:,8)==0 & states_nontarget_sort(:,9)==1);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==1 & states_nontarget_sort(:,7)==0...
  & states_nontarget_sort(:,8)==1 & states_nontarget_sort(:,9)==0);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==1 & states_nontarget_sort(:,7)==0...
  & states_nontarget_sort(:,8)==1 & states_nontarget_sort(:,9)==1);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==1 & states_nontarget_sort(:,7)==1...
  & states_nontarget_sort(:,8)==0 & states_nontarget_sort(:,9)==0);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==1 & states_nontarget_sort(:,7)==1...
  & states_nontarget_sort(:,8)==0 & states_nontarget_sort(:,9)==1);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
```

```
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==1 & states_nontarget_sort(:,7)==1...
  & states_nontarget_sort(:,8)==1 & states_nontarget_sort(:,9)==0);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)];
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))];
r=find(states_nontarget_sort(:,6)==1 & states_nontarget_sort(:,7)==1...
  & states_nontarget_sort(:,8)==1 & states_nontarget_sort(:,9)==1);
StatesNonTargetMid = [StatesNonTargetMid;states_nontarget_sort(r,:)];
NonTargetStateSamples = [NonTargetStateSamples,length(r)]
FalseDetections = [FalseDetections;sum(states_nontarget_sort(r,5))]

% End of grouping and sorting section
% END grouping nontarget data
%  ------------------------------------------------------------------------



% --------------------------------------------------------------------------
% Begin Error statistics - total
fprintf(1,'\n Total MD: %5.4f for %d Targets \n',...
    1-mean(StatesTargetMid(:,5)), target_length);

fprintf(1,'\n Total FA: %5.4f for %d NonTargets \n',...
    mean(StatesNonTargetMid(:,5)), nontarget_length);
%  End Error statistics - total
% -------------------------------------------------------------------------
% --------------------------------------------------------------------------
%  BEGIN 16 state operator error model from available input data.
%    Use target Hi - Lo data for 4 Independent variables.
%    Create operator model data preliminary matrix =
%                        [GSD AA Dwell IET #NT #FA Pfa2 #T #Hits Pmd2]
%OpModelDataP(:,1)=[min(states_target_sort(:,1))*ones(8,1);...
%                         max(states_target_sort(:,1))*ones(8,1)];
OpModelDataP(:,1)=[0.190905*ones(8,1);0.275752*ones(8,1)];
OpModelDataP(:,2)=[45*ones(4,1);90*ones(4,1);45*ones(4,1);90*ones(4,1)];
OpModelDataP(:,3)=[2.240;2.240;4.550;4.550;2.240;2.240;4.550;4.550;...
                   2.240;2.240;4.550;4.550;2.240;2.240;4.550;4.550];
% OpModelDataP(:,3)=[min(states_target_sort(:,3));min(states_target_sort(:,3));...
%          max(states_target_sort(:,3));max(states_target_sort(:,3));...
%          min(states_target_sort(:,3));min(states_target_sort(:,3));...
%          max(states_target_sort(:,3));max(states_target_sort(:,3));...
%          min(states_target_sort(:,3));min(states_target_sort(:,3));...
%          max(states_target_sort(:,3));max(states_target_sort(:,3));...
%          min(states_target_sort(:,3));min(states_target_sort(:,3));...
%          max(states_target_sort(:,3));max(states_target_sort(:,3));...
%          ];
OpModelDataP(:,4)=[0;30;0;30;0;30;0;30;0;30;0;30;0;30;0;30];
OpModelDataP(:,5)= NonTargetStateSamples';
OpModelDataP(:,6)= FalseDetections;
OpModelDataP(:,7)= FalseDetections./NonTargetStateSamples';
OpModelDataP(:,8)= TargetStateSamples';

OpModelDataP(:,9)= DetectionHits;

OpModelDataP(:,10)= ones(16,1) - DetectionHits./TargetStateSamples';


% This is to fix 1 state that has a zero entry.

%OpModelDataP(8,7) = 0.0125;


OpModelDataPR = [OpModelDataP(:,1:4),OpModelDataP(:,7),OpModelDataP(:,10)];

%xlswrite('OpModelDataF86',OpModelDataPR);

%xlswrite('OpModelDataF88',OpModelDataPR);

%xlswrite('OpModelDataF0828',OpModelDataPR);

%%xlswrite('OpModelDataF0920',OpModelDataPR);

% END 16 state operator error model from available input data.

% ----------------------------------------------------------------------




%-------------------------------------------------------------------

% BEGIN interpolation of operator error model from 16 states to 256 states.

% This gives the grid form for the 16 states.

[GSD16,AA16,Dwell16,IET16]=ndgrid(OpModelDataP(1,1):OpModelDataP(16,1)...

   -OpModelDataP(1,1):OpModelDataP(16,1),45:45:90,...

    OpModelDataP(1,3):OpModelDataP(16,3)...

   -OpModelDataP(1,3):OpModelDataP(16,3),0:30:30);

%  This gives the value of the table (Pfa16) in grid form.

Pfa16(:,:,1,1)=[OpModelDataP(1,7),OpModelDataP(5,7);...

                OpModelDataP(9,7),OpModelDataP(13,7)];
```

```
Pfa16(:,:,2,1)=[OpModelDataP(3,7),OpModelDataP(7,7);...
                OpModelDataP(11,7),OpModelDataP(15,7)];
Pfa16(:,:,1,2)=[OpModelDataP(2,7),OpModelDataP(6,7);...
                OpModelDataP(10,7),OpModelDataP(14,7)];
Pfa16(:,:,2,2)=[OpModelDataP(4,7),OpModelDataP(8,7);...
                OpModelDataP(12,7),OpModelDataP(16,7)];

%  This gives the value of the table (Pmd16) in grid form.
Pmd16(:,:,1,1)=[OpModelDataP(1,10),OpModelDataP(5,10);...
                OpModelDataP(9,10),OpModelDataP(13,10)];
Pmd16(:,:,2,1)=[OpModelDataP(3,10),OpModelDataP(7,10);...
                OpModelDataP(11,10),OpModelDataP(15,10)];
Pmd16(:,:,1,2)=[OpModelDataP(2,10),OpModelDataP(6,10);...
                OpModelDataP(10,10),OpModelDataP(14,10)];
Pmd16(:,:,2,2)=[OpModelDataP(4,10),OpModelDataP(8,10);...
                OpModelDataP(12,10),OpModelDataP(16,10)];

%  This gives the grid form for the 256 interpolated states.
[GSD256,AA256,Dwell256,IET256]=ndgrid(OpModelDataP(1,1):(OpModelDataP(16,1)...
   -OpModelDataP(1,1))/3:OpModelDataP(16,1),45:15:90,...
    OpModelDataP(1,3):(OpModelDataP(16,3)-OpModelDataP(1,3))...
    /3:OpModelDataP(16,3),0:10:30);
%  This gives the value of the table Pfa256 and Pmd256 in grid form.
Pfa256=interpn(GSD16,AA16,Dwell16,IET16,Pfa16,GSD256,AA256,Dwell256,IET256);
Pmd256=interpn(GSD16,AA16,Dwell16,IET16,Pmd16,GSD256,AA256,Dwell256,IET256);
```

```
% This section converts the grid model into op model xml file form of
% increasing frequency of variations from right to left.
OpModelDataP256 = 99*ones(256,6);
m=1;
for i=1:4
    for j=1:4
        for k=1:4
            for l=1:4
                OpModelDataP256(m,:)=[GSD256(i,j,k,l),AA256(i,j,k,l),...
       Dwell256(i,j,k,l),IET256(i,j,k,l),Pfa256(i,j,k,l),Pmd256(i,j,k,l)];
                m=m+1;
            end
        end
    end
end


%  End interpolation of operator error model from 16 states to 256 states.
%-----------------------------------------------------------------


%-----------------------------------------------------------------
% BEGIN xml
%  This line saves the 256 state model as an excel file.
%  The numbers on end mean that the op data used to make the end on M,D.
%xlswrite('OpModelData256F0828',OpModelDataP256);
```

```
%%xlswrite('OpModelData256F0920',OpModelDataP256);


%  This line saves the same data as an xml file for use in VSCS.
%modelPrinter(OpModelDataP256,'OpModelData256F0828.xml');
%%modelPrinter(OpModelDataP256,'OpModelData256F0920.xml');
% END xml
% -------------------------------------------------------------------



% END operator model m file
% ----------------------------------------------------------------------------
```

Appendix H: Connection with Communications & Information Theory

H.1 Derivation of Information Gain

The inspected objects are $X = [T, FT]$, where the true classification is either Target (T) or False Target (FT). The un-assisted operator, or the SC-assisted operator, declares the inspected object to be a Target or False Target, $Y = [T_o, FT_o]$. The inspection operation is modeled as an asymmetric binary communication channel, as shown in Figure H.1.

The prior information or density is p, $P_{MD} = 1 - P_{TR}$, and $P_{FA} = 1 - P_{FTR}$.

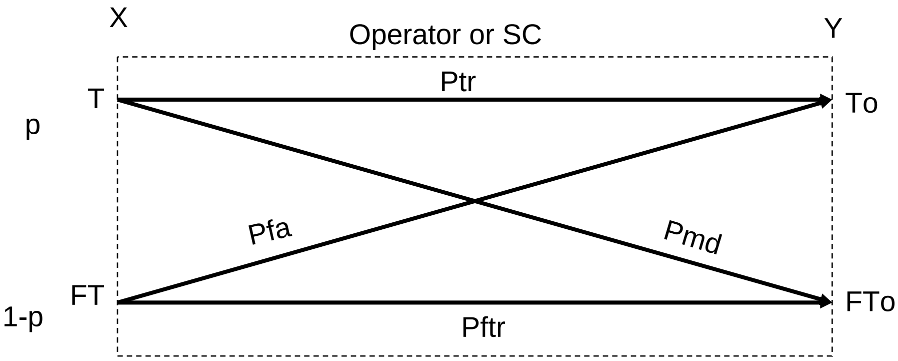


Figure H.1: Binary Communications Channel

Definitions

The Mutual Information between the events $x \in X$ and $y \in Y$ is the information provided about the event x by the occurrence of the event y:

$$I_{X;Y}(x; y) \triangleq \log_2 \frac{P(x|y)}{P(x)} \quad [bits]$$

The Average Mutual Information is:

$$I(X; Y) \triangleq \sum_x \sum_y P(x, y) \log_2 \frac{P(x|y)}{P(x)} \quad [bits]$$

The performance of the operator and/or the SC may be gauged by calculating the information (in bits) provided by them, given the Target density data p and their performance parameters P_{TR} and P_{FTR}. To this end, consider the binary channel shown in Figure H.1 and calculate:

1. The joint probabilities

$$\begin{aligned} P(T, T_o) &= pP_{TR},\ P(FT, FT_o) = (1-p)P_{FTR} \\ P(T, FT_o) &= pP_{MD},\ P(FT, T_o) = (1-p)P_{FA} \end{aligned}$$

2. The probabilities of the events

$$P(T_o) = pP_{TR} + (1-p)P_{FA},\ \ P(FT_o) = (1-p)P_{FTR} + pP_{MD}$$

3. Therefore, the conditional probabilities

$$\begin{aligned} P(T|T_o) &= \frac{pP_{TR}}{pP_{TR} + (1-p)P_{FA}},\ P(FT|FT_o) = \frac{(1-p)P_{FTR}}{(1-p)P_{FTR} + pP_{MD}} \\ P(T|FT_o) &= \frac{pP_{MD}}{(1-p)P_{FTR} + pP_{MD}},\ P(FT|T_o) = \frac{(1-p)P_{FA}}{pP_{TR} + (1-p)P_{FA}} \end{aligned}$$

4. The mutual information

$$\begin{aligned} I(T; T_o) &= \log_2 \frac{P_{TR}}{pP_{TR} + (1-p)P_{FA}} \\ I(FT; FT_o) &= \log_2 \frac{P_{FTR}}{(1-p)P_{FTR} + pP_{MD}} \\ I(T; FT_o) &= \log_2 \frac{P_{MD}}{(1-p)P_{FTR} + pP_{MD}} \\ I(FT; T_o) &= \log_2 \frac{P_{FA}}{pP_{TR} + (1-p)P_{FA}} \end{aligned}$$

5. Average mutual information

$$\begin{aligned} I(X; Y) &= P(T, T_o) \cdot I(T; T_o) + P(FT, FT_o) \cdot I(FT; FT_o) \\ &+ P(T, FT_o) \cdot I(T; FT_o) + P(FT, T_o) \cdot I(FT; T_o) \end{aligned}$$

Hence,

$$\begin{aligned} I(X;Y) &= pP_{TR}\log_2 \frac{P_{TR}}{pP_{TR}+(1-p)P_{FA}} + (1-p)P_{FTR}\log_2 \frac{P_{FTR}}{(1-p)P_{FTR}+pP_{MD}} \\ &+ pP_{MD}\log_2 \frac{P_{MD}}{(1-p)P_{FTR}+pP_{MD}} + (1-p)P_{FA}\log_2 \frac{P_{FA}}{pP_{TR}+(1-p)P_{FA}} \end{aligned} \tag{H.1}$$

Using the mutual information concept to gauge the operator's or SC's performance allows us to bring into the picture the target density parameter p, as opposed to the $PC \overset{\Delta}{=} (P_{TR}+P_{FTR})/2$ performance metric where p plays no role.

Concerning [H.1]:

The expected/average information gain $I(X;Y)$ is a convex function of P_{TR} and P_{FTR} for a fixed "target density" p.

It is interesting to ascertain the channel's parameters P^*_{TR} and P^*_{FTR} where the expected information gain function $I(X;Y)$ attains its minimum for a fixed "target density" p. To this end, differentiate [H.1] in P_{TR} and P_{FTR} and set the derivatives to 0, thus obtaining P^*_{TR} and P^*_{FTR}.

For example, when $p = 1/2$ consider [H.2].

If $p = 1/2$, the average information gain is

$$\begin{aligned} I(X;Y) &= 1 + \frac{1}{2}[P_{TR}\log_2 P_{TR} + P_{FTR}\log_2 P_{FTR} + P_{MD}\log_2 P_{MD} + P_{FA}\log_2 P_{FA} \\ &- (P_{TR}+P_{FA})\log_2(P_{TR}+P_{FA}) - (P_{FTR}+P_{MD})\log_2(P_{FTR}+P_{MD})]\ [bits] \end{aligned} \tag{H.2}$$

Differentiating in P_{TR} yields

$$\log P_{TR} + 1 - \log(1-P_{TR}) - 1 = 0$$

and differentiating in P_{FTR} yields

$$\begin{aligned} \log P_{FTR} + 1 - \log(1 - P_{FTR}) - 1 = 0 \\ \Rightarrow\ \log\frac{P_{TR}}{1 - P_{TR}} = 0,\ \log\frac{P_{FTR}}{1 - P_{FTR}} = 0 \\ \Rightarrow\ P^*_{TR} = 1/2,\ P^*_{FTR} = 1/2 \end{aligned}$$

and we calculate the expected information gain when this channel is used

$$I(X;Y) = 0.$$

Proposition For a given "target density" p, the expected information gain is minimal when the channel's parameters are $P^*_{TR} = p$, $P^*_{FTR} = 1 - p$ (and consequently, $P^*_{MD} = 1 - p$, $P^*_{FA} = p$)—The expected information gain $I(X;Y) = 0$.

Proof Always $I(X;Y) \geq 0$. We calculate

$$\begin{aligned} I(X;Y) &= p^2 \log\frac{p}{p^2 + (1-p)p} + (1-p)^2 \log\frac{1-p}{(1-p)^2 + (1-p)p} \\ &+ (1-p)p \log\frac{1-p}{(1-p)^2 + (1-p)p} + (1-p)p \log\frac{p}{p^2 + (1-p)p} \\ &= p^2 \log 1 + (1-p)^2 \log 1 + p(1-p) \log 1 + (1-p)p \log 1 \\ &= 0 \end{aligned}$$

□

Note: If the operator knows the target density p and he pursues a strategy of flipping a loaded coin, then the expected information gain provided by this operator is 0!

From [H.1] it also follows that if $P_{TR} = P_{FTR} = 1/2$, $I(X;Y) = 0\ \forall p$. Thus, the channel's capacity is $C = 0$! Thus, if the operator pursues a strategy of flipping a fair coin, then the operator conveys no information about the inspected object.

H.2 Justification of the Average Mutual Information Criterion

The entropy[1] $H(X)$ of a discrete random variable X is

$$H(X) \overset{\Delta}{=} -\sum_x p(x) \log_2(p(x))$$

This is the average information gain when told the outcome realization of the random variable X.

Thus, for $X = [T, FT]$

$$\begin{aligned} H(X) &= -p \log_2 p - (1-p) \log_2(1-p) \\ &\overset{\Delta}{=} H(p) \end{aligned}$$

The entropy quantifies our lack of information about the random variable X.

It is in fact the information required to specify x. See Figure H.2.

The conditional entropy

$$H(X|Y) = -\sum_x \sum_y P(x,y) \log P(x|y)$$

This is the average information (the average is taken over x and y) required to specify x after y is known.

In our case, $X = [T, FT],\ Y = [T_o, FT_o]$ and the conditional entropy

$$\begin{aligned} H(X|Y) &= -P(T,T_o) \log_2 P(T|T_o) - P(T,FT_o) \log_2 P(T|FT_o) \\ &- P(FT,T_o) \log_2 P(FT|T_o) - P(FT,FT_o) \log_2 P(FT|FT_o) \end{aligned}$$

[1] Entropy: Assume that you know the pdf but you don't know the result of a draw; however, someone else does. The minimal number of yes/no questions on average you'll need to ask him to find out the result of the draw is $\leq H(X) + 1$

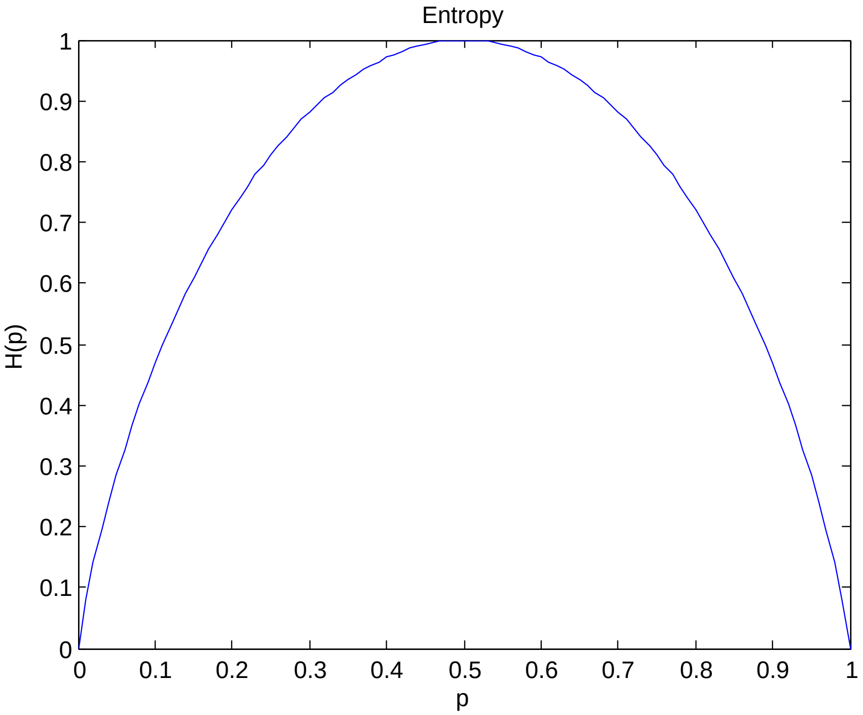


Figure H.2: Entropy as function of density

We then calculate

$$\begin{aligned}
H(X|Y) &= -pP_{TR}\log_2 \frac{pP_{TR}}{pP_{TR}+(1-p)P_{FA}} - (1-p)P_{FTR}\log_2 \frac{(1-p)P_{FTR}}{(1-p)P_{FTR}+pP_{MD}} \\
&- pP_{MD}\log_2 \frac{pP_{MD}}{(1-p)P_{FTR}+pP_{MD}} - (1-p)P_{FA}\log_2 \frac{(1-p)P_{FA}}{pP_{TR}+(1-p)P_{FA}} \\
&= -pP_{TR}\log_2 p - (1-p)P_{FTR}\log_2(1-p) - pP_{MD}\log_2 p - (1-p)P_{FA}\log_2(1-p) - I(X;Y) \\
&= -p\log_2 p - (1-p)\log_2(1-p) - I(X;Y) \\
&= H(X) - I(X;Y)
\end{aligned}$$

Hence,

$$I(X;Y)_a = H(X)_b - H(X|Y)_c \tag{H.3}$$

a: Average information gain due to the knowledge of y.

b: The information required to specify x.

c: The information required to specify x after y is known.

Thus, the average mutual information between x and y is the difference between the entropy of X and the average conditional entropy of X given Y. $I(X;Y)$ is the average amount of uncertainty in X resolved by the observation of the outcome in the Y ensemble: The mutual information is the reduction in uncertainty of X due to the knowledge of Y. The equation [H.3] always holds, not just in the present scenario.

Concerning the average mutual information in the special case where the target density $p = 1/2$, namely equation [H.1]:

$$I(X;Y) = 1 + \frac{1}{2}[H(Y) - H(X,Y)]$$

where the entropy

$$H(X,Y) = -\sum_x \sum_y p(x,y) \log p(x|y)$$

and due to the chain rule of information theory

$$I(X;Y) = 1 - \frac{1}{2}H(X|Y) \quad (< 1, always) \tag{H.4}$$

where the conditional entropy

$$H(X|Y) = -\sum_x \sum_y p(x,y) \log_2 \frac{p(y)}{p(x,y)}.$$

Bottom Line

To evaluate the benefit afforded by the SC,

1. Evaluate the average mutual information according to equation [H.1], using the P_{TR} and P_{FTR} parameters of the operator, *and* the target density p (= 1/2). We obtain $I(X;Y)_o$.

2. Evaluate the average mutual information according to equation [H.1], using the P_{TR} and P_{FTR} parameters of the SC, *and* the target density p (= 1/2). We obtain $I(X;Y)_{SC}$.

3. Calculate

$$\Delta I(X;Y) \overset{\Delta}{=} I(X;Y)_{SC} - I(X;Y)_o \ [bits]$$

This is a replacement for the PC metric and also takes into account the target density p. Can also project ahead to account for different target densities, assuming that the operator's performance (P_{TR} and P_{FTR}) is not affected by the "knowledge" of p.

H.3 Correct Definition of PC

Here, $X = [T, FT]$.

The Percent Correct (PC) is the probability of a correct transmission of a T or FT symbol being sent through the binary communication channel shown in Figure 1. We calculate

$$PC = pP_{TR} + (1-p)P_{FTR}$$

By the same token, sending one symbol through the channel has a Probability of Error (PE) of

$$PE = pP_{MD} + (1-p)P_{FA}$$

Since $P_{MD} = 1 - P_{TR}$ and $P_{FA} = 1 - P_{FTR}$, then $PC + PE = 1$, as expected.

Remark If $p = 1/2$ then indeed $PC = (P_{TR} + P_{FTR})/2$.

In general, see Figure H.3

The PC (and PE) performance metric allows us to calculate the probability P of correctly classifying m objects when $n \geq m$ objects are being inspected. The number of correctly classified objects is a Bernoulli random variable and consequently

$$P = \binom{n}{m}(PC)^m(PE)^{n-m}$$

Note If $0 < p \ll 1$ and, knowing this, the operator blindly adopts the strategy of classifying everything as a FT, that is, $P_{TR} = 0$, and $P_{FTR} = 1$, then $PC = 1 - p \approx 1$ and $PE = p \approx 0$. This creates an illusion of good performance. However, the information gain $I(X;Y) = 0$! This tells us that information gain might be a better performance metric.

As an example, take a $p = 0.05$ target density case where $PC = pP_{TR} + (1-p)P_{FTR}$: Op: FA=1%, MD=70%, PCavg=65%, whereas $PC = 0.05 * 0.3 + 0.95 * 0.99 = 95\%$

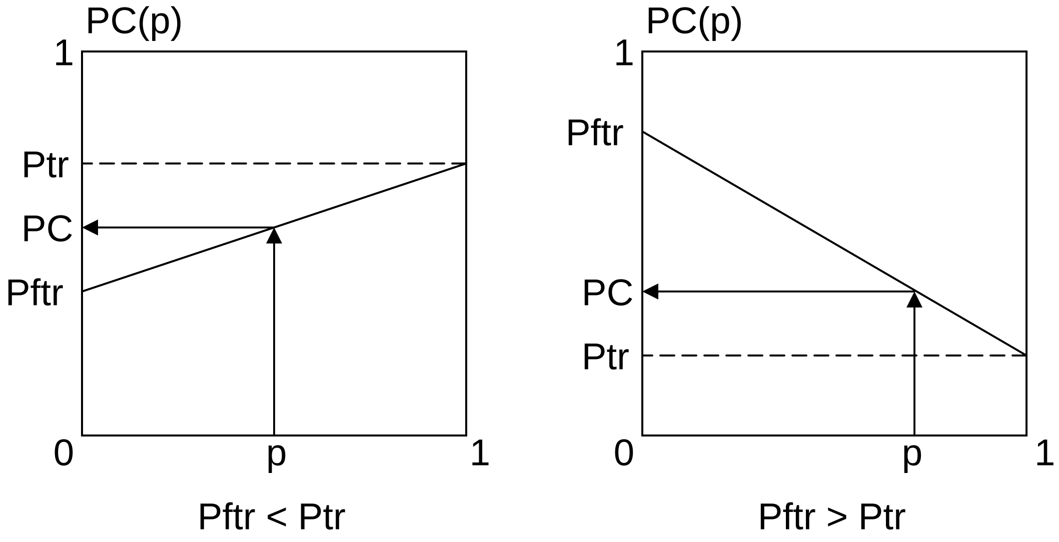


Figure H.3: PC as a function of p

SC: FA=0%, MD=20%, PCavg=90%, whereas $PC = 0.05 * 0.8 + 0.95 * 1.0 = 99\%$

However, the $I(X;Y) \equiv$ Expected Information Gain performance metric:

$$\begin{aligned}
O_p : I(X;Y) &= 0.05 * 0.3 \log_2 \frac{0.3}{0.05 * 0.3 + 0.95 * 0.01} + 0.95 * 0.99 \log_2 \frac{0.99}{0.95 * 0.99 + 0.05 * 0.7} \\
&+ 0.05 * 0.7 \log_2 \frac{0.7}{0.95 * 0.99 + 0.05 * 0.7} + 0.95 * 0.01 \log_2 \frac{0.01}{0.05 * 0.3 + 0.95 * 0.01} \\
&= 0.0451928 \; [bits] \\
SC : I(X;Y) &= 0.05 * 0.8 \log_2 \frac{0.8}{0.05 * 0.8 + 0.95 * 0.0} + 0.95 * 1.0 \log_2 \frac{1.0}{0.95 * 1.0 + 0.05 * 0.2} \\
&+ 0.05 * 0.2 \log_2 \frac{0.2}{0.95 * 1.0 + 0.05 * 0.2} + 0.95 * 0.0 \log_2 \frac{0.0}{0.05 * 0.8 + 0.95 * 0.0} \\
&= 0.206196 \; [bits]
\end{aligned}$$

This needs to be compared to the channel's capacity, C_{O_p} and C_{SC}.

Note: When the operator adopts the strategy of declaring all inspected objects FT ($P_{FTR} = 1$ and $P_{TR} = 0$), then $I(X;Y) = 0$. If the operator uses a p loaded coin, then $I(X;Y) = 0$ also.

H.4 The Capacity of the Communication Channel Metric

$$C \overset{\Delta}{=} \max_{p} I(X;Y)\ [bits]$$

It is convenient to write

$$\begin{aligned} I(X;Y) &= -pP_{TR}\log(p+\frac{P_{FA}}{P_{TR}}(1-p))+(1-p)P_{FTR}\log(\frac{P_{MD}}{P_{FTR}}p+1-p) \\ &- pP_{MD}\log(p+\frac{P_{FTR}}{P_{MD}}(1-p))+(p-1)P_{FA}\log(1-p+\frac{P_{MD}}{P_{FTR}}p) \end{aligned}$$

Differentiate in p:

$$\begin{aligned} -P_{TR}\log(p+\frac{P_{FA}}{P_{TR}}(1-p))-pP_{TR}\frac{1-\frac{P_{FA}}{P_{TR}}}{p+\frac{P_{FA}}{P_{TR}}(1-p)} \\ +P_{FTR}\log(1-p+\frac{P_{MD}}{P_{FTR}}p)+(1-p)P_{FTR}\frac{\frac{P_{MD}}{P_{FTR}}-1}{1-p+\frac{P_{MD}}{P_{FTR}}p} \\ -P_{MD}\log(p+\frac{P_{FTR}}{P_{MD}}(1-p))-pP_{MD}\frac{1-\frac{P_{FTR}}{P_{MD}}}{p+\frac{P_{FTR}}{P_{MD}}(1-p)} \\ +P_{FA}\log(1-p+\frac{P_{TR}}{P_{FA}}p)+(p-1)P_{FA}\frac{\frac{P_{TR}}{P_{FA}}-1}{1-p+\frac{P_{TR}}{P_{FA}}p} = 0 \end{aligned}$$

=>

$$\begin{aligned} -pP_{TR}\frac{1}{\frac{P_{FA}}{P_{TR}-P_{FA}}+p}+(p-1)P_{FTR}\frac{1}{\frac{P_{FTR}}{P_{MD}-P_{FTR}}+p} \\ -pP_{MD}\frac{1}{\frac{P_{FTR}}{P_{MD}-P_{FTR}}+p}+(p-1)P_{FA}\frac{1}{\frac{P_{FA}}{P_{TR}-P_{FA}}+p} \\ +\log\frac{(1-p+\frac{P_{MD}}{P_{FTR}}p)^{P_{FTR}}(1-p+\frac{P_{TR}}{P_{FA}}p)^{P_{FA}}}{[p+\frac{P_{FA}}{P_{TR}}(1-p)]^{P_{TR}}[p+\frac{P_{FTR}}{P_{MD}}(1-p)]^{P_{MD}}} = 0 \end{aligned}$$

=>

$$\begin{aligned} \frac{(P_{FA}-P_{TR})p-P_{FA}}{\frac{P_{FA}}{P_{TR}-P_{FA}}+p}+\frac{(P_{FTR}-P_{MD})p-P_{FTR}}{\frac{P_{FTR}}{P_{MD}-P_{FTR}}+p} \\ +\log\frac{(1-p+\frac{P_{MD}}{P_{FTR}}p)^{P_{FTR}}(1-p+\frac{P_{TR}}{P_{FA}}p)^{P_{FA}}}{[p+\frac{P_{FA}}{P_{TR}}(1-p)]^{P_{TR}}[p+\frac{P_{FTR}}{P_{MD}}(1-p)]^{P_{MD}}} = 0 \end{aligned}$$

=>

$$(1-p+\frac{P_{MD}}{P_{FTR}}p)^{P_{FTR}}(1-p+\frac{P_{TR}}{P_{FA}}p)^{P_{FA}} = [p+\frac{P_{FA}}{P_{TR}}(1-p)]^{P_{TR}}[p+\frac{P_{FTR}}{P_{MD}}(1-p)]^{P_{MD}}$$

=>

$$[1+(\frac{1-P_{TR}}{P_{FTR}}-1)p]^{P_{FTR}}[1+(\frac{P_{TR}}{P_{FA}}-1)p]^{P_{FA}} = [\frac{P_{FA}}{P_{TR}}+(1-\frac{P_{FA}}{P_{TR}})p]^{P_{TR}}[\frac{P_{FTR}}{P_{MD}}+(1-\frac{P_{FTR}}{P_{MD}})p]^{P_{MD}}$$

=>

$$[1+(\frac{1-P_{TR}}{P_{FTR}}-1)p]^{P_{FTR}}[1+(1+\frac{P_{TR}}{P_{FTR}}-1)p]^{1-P_{FTR}} = [\frac{1-P_{FTR}}{P_{TR}}+(1-\frac{1-P_{FTR}}{P_{TR}})p]^{P_{TR}}[\frac{P_{FTR}}{1-P_{TR}}+(1-\frac{P_{FTR}}{1-P_{TR}})p]^{1-P} \tag{H.5}$$

Must solve above equation in p.

Example: *Symmetric Binary Channel:* $P_{FTR} = P_{TR}$

Equation H.5 is now

$$\left[1+(\frac{1}{P_{TR}}-2)p\right]^{P_{TR}}\left[1+(\frac{P_{TR}}{1-P_{TR}}-1)p\right]^{1-P_{TR}} = \left[\frac{1}{P_{TR}}-1+(2-\frac{1}{P_{TR}})p\right]^{P_{TR}}\left[\frac{P_{TR}}{1-P_{TR}}+(1-\frac{P_{TR}}{1-P_{TR}})p\right]^{1-P_{TR}}$$

=>

$$\left[\frac{P_{TR}+(1-2P_{TR})p}{1-P_{TR}+(2P_{TR}-1)p}\right]^{P_{TR}} = \left[\frac{P_{TR}+(1-2P_{TR})p}{1-P_{TR}+(2P_{TR}-1)p}\right]^{1-P_{TR}}$$

=>

$$1 = \frac{P_{TR}+(1-2P_{TR})p}{1-P_{TR}+(2P_{TR}-1)p}$$

=>

$$\left[\frac{1-P_{TR}+(2P_{TR}-1)p}{1-P_{TR}+(2P_{TR}-1)p} * \frac{P_{TR}+(1-2P_{TR})p}{P_{TR}+(1-2P_{TR})p}\right]^{P_{TR}} = \frac{P_{TR}+(1-2P_{TR})p}{1-P_{TR}+(2P_{TR}-1)p}$$

=>

$$1 - P_{TR} + (2P_{TR} - 1)p = P_{TR} + (1 - 2P_{TR})p$$

=>

$$2(1 - 2P_{TR})p = 1 - 2P_{TR}$$

Hence, when $P_{FTR} = P_{TR}$, $p^* = 1/2$ and

$$C^* = 1 + P_{TR}\log_2 P_{TR} + (1 - P_{TR})\log_2(1 - P_{TR}) \ < \ 1 \ [bits]$$

provided that $P_{TR} \neq 1/2$. If $P_{TR} = P_{FTR} = 1/2$, $I(X;Y) = 0 \ \forall p$ so that $C = 0$.

Concerning equation H.5:

The Average Mutual Information function is concave in p. Therefore, setting its derivative equal to zero yields yields p^* and the Capacity of the communications channel is easily calculated.

Equation H.5 is manipulated as follows.

$$\begin{aligned}[\frac{P_{FTR} + (1 - P_{TR} - P_{FTR})p}{1 - P_{FTR} - (1 - P_{TR} - P_{FTR})p} * \frac{1 - P_{FTR}}{P_{FTR}}]^{P_{FTR}} \\ * \frac{1 - P_{FTR} - (1 - P_{TR} - P_{FTR})p}{P_{FTR} + (1 - P_{TR} - P_{FTR})p} * \frac{1 - P_{TR}}{1 - P_{FTR}} \\ = [\frac{1 - P_{FTR} - (1 - P_{TR} - P_{FTR})p}{P_{FTR} + (1 - P_{TR} - P_{FTR})p} * \frac{1 - P_{TR}}{P_{TR}}]^{P_{TR}}\end{aligned}$$

Let $x \overset{\Delta}{=} (1 - P_{TR} - P_{FTR})p$

Thus,

$$\begin{aligned}(\frac{P_{FTR} + x}{1 - P_{FTR} - x})^{P_{FTR}}(\frac{1 - P_{FTR}}{P_{FTR}})^{P_{FTR}} * \frac{1 - P_{FTR} - x}{P_{FTR} + x} * \frac{1 - P_{TR}}{1 - P_{FTR}} \\ = (\frac{1 - P_{FTR} - x}{P_{FTR} + x})^{P_{TR}}(\frac{1 - P_{TR}}{P_{FTR}})^{P_{TR}}\end{aligned}$$

=>

$$(\frac{P_{FTR}+x}{1-P_{FTR}-x})^{P_{TR}+P_{FTR}-1} = \frac{P_{FTR}^{P_{FTR}}}{P_{TR}^{P_{TR}}} * \frac{(1-P_{FTR})^{1-P_{FTR}}}{(1-P_{TR})^{1-P_{TR}}}$$

Redefine $x \triangleq (1-P_{TR}-P_{FTR})p + P_{FTR}$

=>

$$\frac{x}{1-x} = \left[\frac{P_{FTR}^{P_{FTR}}}{P_{TR}^{P_{TR}}} * \frac{(1-P_{FTR})^{1-P_{FTR}}}{(1-P_{TR})^{1-P_{TR}}}\right]^{\frac{1}{P_{TR}+P_{FTR}-1}}$$

=>

$$\frac{1-x}{x} = \left[\frac{P_{FTR}^{P_{FTR}}}{P_{TR}^{P_{TR}}} * \frac{(1-P_{FTR})^{1-P_{FTR}}}{(1-P_{TR})^{1-P_{TR}}}\right]^{\frac{1}{1-P_{TR}-P_{FTR}}}$$

=>

$$x = \frac{1}{1+\left[\frac{P_{FTR}^{P_{FTR}}}{P_{TR}^{P_{TR}}} * \frac{(1-P_{FTR})^{1-P_{FTR}}}{(1-P_{TR})^{1-P_{TR}}}\right]^{\frac{1}{1-P_{TR}-P_{FTR}}}}$$

=>

$$p^* = \frac{1}{1-P_{TR}-P_{FTR}} * \left\{\frac{1}{1+\left[\frac{P_{FTR}^{P_{FTR}}}{P_{TR}^{P_{TR}}} * \frac{(1-P_{FTR})^{1-P_{FTR}}}{(1-P_{TR})^{1-P_{TR}}}\right]^{\frac{1}{1-P_{TR}-P_{FTR}}}} - P_{FTR}\right\} \qquad \text{(H.6)}$$

Inserting the expression for p^* by equation [H.6] into equation [5.19] yields the "communication's channel" capacity $C = C(P_{TR}, P_{FTR})$.

Corollary In the special case of a symmetric binary channel where $P_{TR} = P_{FTR},\ p^* = 1/2$ and the capacity of the channel is

$$C = 1 + P_{TR}\log_2 P_{TR} + (1-P_{TR})\log_2(1-P_{TR}) \ < \ 1\ [bits],$$

provided that $P_{TR} \neq 1/2$. If $P_{TR} = P_{FTR} = 1/2,\ I(X;Y) = 0\ \forall\ p$ so that $C = 0$.

□

Data for density $p = 0.5$

Op: $P_{FA} = 0.35,\ P_{MD} = 0.17$

SC: $P_{FA} = 0.05,\ P_{MD} = 0.05.$

Note: $P_{TR} = P_{FTR}$ => $p^* = 1/2$

=> $C^* = 1 + 0.95 \log_2 0.95 + 0.05 \log_2 0.05$ [*bits*].

For $p = 0.05$

Op: $P_{FA} = 0.0,\ P_{MD} = 0.5$

Note: On a second visit, which is always under optimal conditions, the operator's $P^{(2)}_{MD} \approx 0,\ P^{(2)}_{FA} \approx 0.$

Remark: The formula for p^*, which yields the channel's capacity, can also be written as

$$p^* = \frac{1}{1 - P_{MD} - P_{FA}} * \left\{ 1 - P_{FA} - \frac{1}{1 + \left[\frac{P_{FA}^{P_{TR}}}{P_{FTR}^{P_{FTR}}} * \frac{P_{MD}^{P_{MD}}}{P_{FA}^{P_{FA}}} \right]^{\frac{1}{1-P_{MD}-P_{FA}}}} \right\}$$

Note: When $P_{MD} = 0$ and $P_{FA} = 0$, $p^* = 1/2$ and then $C^* = 1$ as expected.

In general, given the operator or SC data P_{TR} and P_{FTR}, can calculate p^* and thus can calculate the channel's capacity C; we here contend that the operator or SC act as a communication channel. We have modeled the operator and/or SC as a binary communication channel so that the natural performance metric is its capacity

$$C = \max_p I(X; Y) \ [bits]$$

Thus, $C = C(P_{TR}, P_{FTR})$ and is no longer dependent on the prior information/target density parameter p. Hence, calculate $C(P_{TR_{Op}}, P_{FTR_{Op}})$ and $C(P_{TR_{SC}}, P_{FTR_{SC}})$ and show $C(P_{TR_{SC}}, P_{FTR_{SC}}) > C(P_{TR_{Op}}, P_{FTR_{Op}})$.

The capacity of a communication channel C, is the maximum rate of errorless information transmission over the channel. Here "rate" means number of bits/symbols (T or FT) per transmission. Since we want to correctly transmit/classify one bit, where a bit stands for a T or FT, our channel whose capacity $C < 1$ [bit] is not up to the task. In order to use such a channel whose capacity $C < 1$ [bit], it is standard practice in the field of communications to send the same bit repeatedly through the channel, that is, in our case, re-inspect the object. One then makes a determination according to a voting scheme where the majority prevails. It is of course assumed that the operator's parameters P_{TR} and P_{FTR} are stationary—not so in our case—and, most importantly, the operator's determinations are statistically independent.

How many times n would one need to retransmit the one bit of information: $n = 1/C$ and the data *rate* is $1/n$ $(= C)$.

Obviously, n needs to be an odd number in order to be able to employ a voting scheme.

The probability of error is then

$$P((n+1)/2 \text{ or greater incorrect classifications }) = \sum_{j=(n+1)/2} \binom{n}{j} (PE)^j (PC)^{n-j}$$

We have not used a voting scheme and majority rule. Instead, a probability threshold P_t was used for classification and $n = 2$. The probability P_2 that the object is a T after the outcome of two inspections has become available is calculated as in the Kalman filtering paradigm.

First and foremost, $n = 2$ is adequate if

$$C(P_{TR_{Op}}, P_{FTR_{Op}}) \geq 0.5 \text{ bits} \qquad \text{or} \qquad C(P_{TR_{SC}}, P_{FTR_{SC}}) \approx 1 \text{ bit}$$

H.5 The Probability of Error when a Probability Threshold P_t is used

Two inspections: $n = 2$. For example, $P_t = 0.5$.

After two inspections, the operator's output is (T_o, T_o), or (T_o, FT_o), or (FT_o, T_o), or (FT_o, FT_o).

Given the operator's output, the probability $P_2(T|\cdot,\cdot)$ that the inspected object is a T is calculated—see equations (5.7, 5.8) where a discrete Kalman filtering recursive algorithm is used to calculate $P_1(T|\cdot)$:

We have the following result.

Theorem

$$\begin{aligned}
P(T|T_o) &= \frac{pP_{TR}}{pP_{TR} + (1-p)(1-P_{FTR})} \\
P(FT|T_o) &= \frac{(1-p)(1-P_{FTR})}{pP_{TR} + (1-p)(1-P_{FTR})} \\
P(FT|FT_o) &= \frac{(1-p)P_{FTR}}{p(1-P_{TR}) + (1-p)P_{FTR}} \\
P(T|FT_o) &= \frac{p(1-P_{TR})}{p(1-P_{TR}) + (1-p)P_{FTR}}
\end{aligned}$$

□

There are two failure modes:

1. The probability of error is the probability that after two inspections of an object which is in fact a target T, the calculated probability that the object is a T,

$P_2(T|\cdot,\cdot) < P_t$, so the object is erroneously classified a FT

or,

2. The probability of error is the probability that after two inspections of an object which is in fact a FT, the calculated probability that the object is a T,

$P_2(T|\cdot,\cdot) > P_t$, so the object is erroneously classified a T.

The calculated probability after two inspections $P_2(T|\cdot,\cdot)$ is a function of the 4 possible determinations: (T_o, T_o), (T_o, FT_o), (FT_o, T_o), or (FT_o, FT_o).

Suppose, for example, the problem parameters p, $P^{(1)}_{TR}$, $P^{(1)}_{FTR}$, $P^{(2)}_{TR}$, $P^{(2)}_{FTR}$ are s.t.

Assume 1:

$P_2(T|T_o, T_o) > P_t$, $P_2(T|T_o, FT_o) < P_t$, $P_2(T|FT_o, T_o) > P_t$, and $P_2(T|FT_o, FT_o) < P_t$. (In other words, the results of the second inspection are more believable.)

An error of the first kind will occur if the inspected object is a T and the operator said (T_o, FT_o) or the operator said (FT_o, FT_o). Now, the probability that the operator will say (T_o, FT_o) when the object is indeed a T, is

$$P_o(T_o, FT_o) = P^{(1)}_{TR} \cdot P^{(2)}_{MD}$$

and the probability that the operator will say (FT_o, FT_o) when the object is in fact a T is:

$$P_o(FT_o, FT_o) = P^{(1)}_{MD} \cdot P^{(2)}_{MD}$$

and so the probability of error in the first failure mode is

$$P_{E_1} = P^{(1)}_{TR} \cdot P^{(2)}_{MD} + P^{(1)}_{MD} \cdot P^{(2)}_{MD} = P^{(2)}_{MD}$$

Failure mode 2 arises when the inspected object is a FT and the calculated probability

$$P_2(T|\cdot, \cdot) > P_t.$$

<u>Assume</u> 2:

$P_2(FT|FT_o, FT_o) < P_t$, $P_2(FT|T_o, FT_o) < P_t$, $P_2(FT|FT_o, T_o) > P_t$, and $P_2(FT|FT_o, FT_o) > P_t$. (Again, we assume the results of the second inspection are more believable.)

According to these assumptions, a mode 2 failure will occur when the operator says (FT_o, T_o) or (T_o, T_o) when in fact the inspected object is a FT.

The probability that the operator will say (FT_o, T_o) when in fact the inspected object is a FT is

$$P_o(FT_o, T_o) = P^{(1)}_{FTR} \cdot P^{(2)}_{FA},$$

and the probability that the operator will say (T_o, T_o) when the object is a FT is

$$P_o(T_o, T_o) = P^{(1)}_{FA} \cdot P^{(2)}_{FA},$$

Hence, the probability of error in case 2 is

$$P_{E_2} = P^{(1)}_{FTR} \cdot P^{(2)}_{FA} + P^{(1)}_{FA} \cdot P^{(2)}_{FA} = P^{(2)}_{FA}$$

Hence, the probability of error in our two inspection scheme is

$$P_E = pP_{E_1} + (1-p)P_{E_2} = pP^{(2)}_{MD} + (1-p)P^{(2)}_{FA}$$

Thus,

$$P_E = pP^{(2)}_{MD} + (1-p)P^{(2)}_{FA}.$$

The second inspection is crucial. The results of the first inspection are immaterial. This is because the “state” for the second time around is considerably better. Thus, the validity of the final formula above for P_E rests on Assumptions 1&2. If, for example,

$P_{MD}^{(2)} \approx 0,\ P_{FA}^{(2)} \Rightarrow P_E \approx 0.$

Consider the symmetric binary channel where $P_{TR}^{(1)} = P_{TR}^{(2)} = 1-\varepsilon,\ P_{FTR}^{(1)} = P_{FTR}^{(2)} = 1-\varepsilon.$

We calculate

$$\begin{aligned}
P_1(T|T_o) &= \frac{p(1-\varepsilon)}{p+\varepsilon-2p\varepsilon}\\
P_1(FT|T_o) &= \frac{(1-p)\varepsilon}{p+\varepsilon-2p\varepsilon}\\
P_1(FT|FT_o) &= \frac{(1-p)(1-\varepsilon)}{1+2p\varepsilon-p-\varepsilon}\\
P_1(T|FT_o) &= \frac{p\varepsilon}{1+2p\varepsilon-p-\varepsilon}
\end{aligned}$$

=> In the case where $n = 2$,

$$P_2(T|T_o,T_o) = \frac{p(1-\varepsilon)^2}{p-2p\varepsilon+\varepsilon^2},\ P_2(T|FT_o,T_o) = p,\ P_2(T|FT_o,T_o) = p\ \forall\varepsilon$$

$$P_2(T|FT_o,FT_o) = \frac{p\varepsilon^2}{1-p+2p\varepsilon+\varepsilon^2\varepsilon}$$

Note If $0 \le \varepsilon < 1/2$,

$P_2(T|T_o,T_o) > p$ and $P_2(T|FT_o,FT_o) < p$

Set the probability threshold $P_t > p$ s.t. $P_2(T|T_o,T_o) > P_t$.

Thus, the situation is $0 < P_2(T|FT_o,FT_o) < p < P_t < P_2(T|T_o,T_o) < 1$.

There are two failure modes: The probability of error P_E is

1. The probability that after two inspections of an object which in fact a T, the calculated probability that the object is a T

$P_2(T|\cdot,\cdot) < P_t$

so that T is misclassified a FT,

or,

2. The probability that after two inspections of an object which is in fact a FT, the calculated probability

$$P_2(T|\cdot,\cdot) > P_t$$

so that FT is misclassified a T.

Failure mode 1 can arise when the operator said

a) (T_o, FT_o). The probability of this happening is $P_{TR}^{(1)} \cdot P_{MD}^{(2)} = (1-\varepsilon)\varepsilon$

b) (FT_o, T_o). The probability of this happening is $P_{MD}^{(1)} \cdot P_{TR}^{(2)} = \varepsilon(1-\varepsilon)$

c) (FT_o, FT_o). The probability of this happening is $P_{MD}^{(1)} \cdot P_{MD}^{(2)} = \varepsilon \cdot \varepsilon$

Hence,

$$P_{E_1} = 2\varepsilon(1-\varepsilon) + \varepsilon^2 = 2\varepsilon - \varepsilon^2$$

Failure mode 2 can only arise when the operator said (T_o, T_o). The probability of this happening is $P_{FA}^{(1)} \cdot P_{FA}^{(2)} = \varepsilon \cdot \varepsilon$

Hence,

$$P_{E_2} = \epsilon^2,$$

and we calculate the probability of failure

$$\begin{aligned} P_E &= pP_{E_1} + (1-p)P_{E_2} \\ &= p(2\varepsilon - \varepsilon^2) + (1-p)\varepsilon^2 \end{aligned}$$

=>

$$P_E = 2p\varepsilon(1-\varepsilon) + \varepsilon^2$$

This result holds, provided that the probability threshold P_t satisfies the inequality

$$p < P_t < P_2(T|T_o, T_o)$$

and $0 \le \varepsilon < 1/2$. Thus, need $0 \le \varepsilon < 1/2$ and

$$p < P_t < \frac{p(1-\varepsilon)^2}{p - 2p\varepsilon + \varepsilon^2}$$

=>

Need

$$0 \leq \varepsilon < 1/2$$

and the following holds.

Proposition Consider the symmetric binary channel where $P^{(1)}_{TR} = P^{(2)}_{TR} = 1 - \varepsilon$ and $P^{(1)}_{FTR} = P^{(2)}_{FTR} = 1 - \varepsilon$. If $0 \leq \varepsilon < 1/2$ and the probability threshold satisfies

$$p < P_t < \frac{p(1-\varepsilon)^2}{p - 2p\varepsilon + \varepsilon^2}$$

then if $n = 2$, the probability of error

$$P_E = 2p\varepsilon(1-\varepsilon) + \varepsilon^2$$

□

Example If $p = 0.5$, need probability threshold $1/2 < P_t < 1/(1 + (\varepsilon/(1-\varepsilon))^2)$ and $P_E = \varepsilon$. If $p = 0.05$, and $0.05 < P_t < 1/(1 + 19(\varepsilon/(1-\varepsilon))^2)$ and $P_E = 0.1\varepsilon + 0.9\varepsilon^2$.

At the same time, the capacity of the symmetric binary channel where $P_{TR} = P_{FTR} = 1 - \varepsilon$, is

$$C = 1 + \varepsilon \log_2 \varepsilon + (1-\varepsilon)\log_2(1-\varepsilon) \qquad \text{[bits](<1).}$$

If, for example, $\varepsilon = 0.01$ then $C = 0.9192069$, $PC = 1 - \varepsilon = 1 - 0.99$, and $PE = \varepsilon = 0.01$. If $n = 3$ and the majority rule is applied, the probability of error

$$\begin{aligned} P_E &= \sum_{j=(n+1)/2}^{n} \binom{n}{j} (PE)^j (PC)^{n-j} \\ &= \binom{3}{2}\varepsilon^2(1-\varepsilon) + \binom{3}{3}\varepsilon^3 = \varepsilon^2(3-2\varepsilon) \\ &= \binom{3}{2} * 0.01^2 * 0.99 + \binom{3}{3} * 0.01 * 1 \\ &= 0.000298 \end{aligned}$$

If however $n = 2$ and the voting rule is s.t. a T is accepted iff the operator says (T_o, T_o), otherwise the object is classified a FT, then a classification error will occur if:

1. The object is a T and the operator said (T_o, FT_o) or (FT_o, T_o), or (FT_o, FT_o). The probability of this occurring is $P_{E_1} = (1 - \varepsilon)\varepsilon + \varepsilon(1 - \varepsilon) + \varepsilon^2$.

2. The object is a FT and the operator said (T_o, T_o). The probability of this occurring is $P_{E_2} = \varepsilon^2$.

Hence,

$$\begin{aligned} P_E &= pP_{E_1} + (1 - p)P_{E_2} \\ &= p[2\varepsilon(1 - \varepsilon) + \varepsilon^2] + (1 - p)\varepsilon^2 \\ P_E &= 2p\varepsilon(1 - \varepsilon) + \varepsilon^2 \end{aligned}$$

as before, when a probability threshold P_t was used.

Consider now the case $n = 3$ and the use of a discrete KF algorithm and probability threshold P_t. We calculate

$$\begin{aligned} &P_3(T|T_o, T_o, T_o) = \tfrac{p(1-\in)^3}{p(1-3\in+3\in^2)-\in^3}, && P_3(T|T_o, T_o, FT_o) = \tfrac{p(1-\in)}{p-2p\in+\in} \\ &P_3(T|T_o, FT_o, T_o) = \tfrac{p(1-\in)}{p-2p\in+\in}, && P_3(T|T_o, FT_o, FT_o) = \tfrac{p\in}{1-p-\in+2p\in} \\ &P_3(T|FT_o, T_o, T_o) = \tfrac{p(1-\in)}{p-2p\in+\in}, && P_3(T|FT_o, T_o, FT_o) = \tfrac{p\in}{1-p-\in+2p\in} \\ &P_3(T|FT_o, FT_o, T_o) = \tfrac{p\in}{1-p-\in+2p\in}, && P_3(T|FT_o, FT_o, FT_o) = \tfrac{p\in^3}{(1-\in)^3+p(2\in^3-3\in^2+3\in-1)} \end{aligned}$$

Assume $0 < \varepsilon < 1/2$ and set $P_t = \frac{p(1-\varepsilon)}{p-2p\varepsilon+\varepsilon}$

Because

$$\begin{aligned} &P_3(T|T_o, T_o, T_o) > P_3(T|T_o, T_o, FT_o) = P_3(T|T_o, FT_o, T_o) = P_3(T|FT_o, T_o, T_o) \\ &> P_3(T|T_o, FT_o, FT_o) = P_3(T|FT_o, T_o, FT_o) = P_3(T|FT_o, FT_o, T_o) \\ &> P_3(T|FT_o, FT_o, FT_o) \end{aligned}$$

Set the probability threshold

$$P_3(T|T_o, FT_o, FT_o) < P_t < P_3(T|T_o, T_o, FT_o).$$

1. An error will occur if the inspected object is a T and the operator said (FT_o, FT_o, FT_o),

or (T_o, FT_o, FT_o), (FT_o, T_o, FT_o), or (FT_o, FT_o, T_o), whereupon the object will be misclassified a FT.

2. An error will occur if the inspected object is a FT and the operator said (T_o, T_o, T_o), (T_o, T_o, FT_o), or (T_o, FT_o, T_o), or (FT_o, T_o, T_o), whereupon the object will be misclassified a T.

The probability of 1. occurring is

$$P_{E_1} = \varepsilon^3 + 3(1-\varepsilon)\varepsilon^2 = 3\varepsilon^2 - 2\varepsilon^3$$

The probability of 2. occurring is

$$P_{E_2} = \varepsilon^3 + 3(1-\varepsilon)\varepsilon^2 = 3\varepsilon^2 - 2\varepsilon^3$$

=> The probability of error

$$P_E = pP_{E_1} + (1-p)P_{E_2}$$

$$P_E = p(3\varepsilon^2 - 2\varepsilon) + (1-p)(3\varepsilon^2 - 2\varepsilon)$$

=> The probability of error

$$P_E = \varepsilon^2(3 - 2\varepsilon)$$

as before, when the majority rule was used.

The Point:

- Using the probability threshold as opposed to the majority rule allows one to have an even number n of inspections,
- Using the probability threshold as opposed to the majority rule allows us to consider the case where $P^{(1)}_{TR} \neq P^{(2)}_{TR} \neq P^{(3)}_{TR}$, and $P^{(1)}_{FTR} \neq P^{(2)}_{FTR} \neq P^{(3)}_{FTR}$.

If $0 \leq \varepsilon < 1/2$, then

$$P_3(T|T_o, T_o, T_o) > P_3(T|T_o, T_o, FT_o) = P_3(T|T_o, FT_o, T_o) = P_3(T|FT_o, T_o, T_o) > p$$

$$P_3(T|FT_o, T_o, T_o) > P_3(T|FT_o, FT_o, T_o) = P_3(T|T_o, FT_o, FT_o) = P_3(T|FT_o, T_o, FT_o) < p$$

In addition,

$$P_3(T|FT_o, FT_o, FT_o) < P_3(T|FT_o, FT_o, T_o)\ (< p)$$

let the probability threshold be set at $P_t = p$.

The probability of error is calculated as follows:

1. The object is a T and the “operator” said (FT_o, FT_o, T_o), or (T_o, FT_o, FT_o), or (FT_o, T_o, FT_o), or (FT_o, FT_o, FT_o).

2. The object is a FT and the “operator” said (T_o, T_o, FT_o), or (T_o, FT_o, T_o), or (FT_o, T_o, T_o), or (T_o, T_o, T_o).

$$P_{E_1} = 3\varepsilon^2(1-\varepsilon) + \varepsilon^3$$

$$P_{E_2} = 3\varepsilon^2(1-\varepsilon) + \varepsilon^3$$

=>

$$P_E = pP_{E_1} + (1-p)P_{E_2}$$

=>

$$P_E = 3\varepsilon^2(1-\varepsilon) + \varepsilon^3 = \varepsilon^2(3-2\varepsilon)$$

When the majority rule is used,

$$P_E = \varepsilon^2(3-2\varepsilon).$$

Same result.

In general, we calculate

$$P(T|T_o, T_o)) = \frac{\frac{pP^{(1)}_{TR}}{pP^{(1)}_{TR}+(1-p)(1-P^{(1)}_{FTR})} * P^{(2)}_{TR}}{\frac{pP^{(1)}_{TR}}{pP^{(1)}_{TR}+(1-p)(1-P^{(1)}_{FTR})} * P^{(2)}_{TR} + (1-P^{(2)}_{FTR})(1-\frac{pP^{(1)}_{TR}}{pP^{(1)}_{TR}+(1-p)(1-P^{(1)}_{FTR})})}$$

=>

$$P(T|T_o, T_o)) = \frac{pP^{(1)}_{TR}P^{(2)}_{TR}}{pP^{(1)}_{TR}P^{(2)}_{TR} + (1-p)(1-P^{(1)}_{FTR})(1-P^{(2)}_{FTR})}$$

$$P(T|FT_o, FT_o)) = \frac{p(1-P^{(1)}_{TR})(1-P^{(2)}_{TR})}{p(1-P^{(1)}_{TR})(1-P^{(2)}_{TR}) + (1-p)P^{(1)}_{FTR}P^{(2)}_{FTR}}$$

$$P(T|FT_o, T_o)) = \frac{p(1-P^{(1)}_{TR})P^{(2)}_{TR}}{p(1-P^{(1)}_{TR})P^{(2)}_{TR} + (1-p)P^{(1)}_{FTR}(1-P^{(2)}_{FTR})}$$

$$P(T|T_o, FT_o)) = \frac{pP^{(1)}_{TR}(1-P^{(2)}_{TR})}{pP^{(1)}_{TR}(1-P^{(2)}_{TR}) + (1-p)(1-P^{(1)}_{FTR})P^{(2)}_{FTR}}$$

Now

$$P(T|FT_o, FT_o) < P(T|FT_o, T_o) < P(T|T_o, T_o)$$

and

$$P(T|FT_o, FT_o) < P(T|T_o, FT_o) < P(T|T_o, T_o).$$

Question: Is $P(T|T_o, FT_o) < P(T|FT_o, T_o)$, or, is $P(T|T_o, FT_o) > P(T|FT_o, T_o)$?

We'll assume

$$P(T|T_o, FT_o) < P(T|FT_o, T_o)$$

($\equiv$ the second look is a better look).

Hence,

$P(T|FT_o, FT_o) < P(T|T_o, FT_o) < P(T|FT_o, T_o) < P(T|T_o, T_o)$

Where does one fit in the probability threshold P_t?

But first, where does the target density p fit in?

The following holds:

$P(T|FT_o, FT_o) < p \quad iff \quad (1 - P_{TR}^{(1)})(1 - P_{TR}^{(2)}) < P_{FTR}^{(1)} P_{FTR}^{(2)}$

$P(T|T_o, T_o) > p \quad iff \quad P_{TR}^{(1)} P_{TR}^{(2)} > (1 - P_{FTR}^{(1)})(1 - P_{FTR}^{(2)})$

=>

$P(T|FT_o, FT_o) < p \quad iff \quad P_{FA}^{(1)} + P_{FA}^{(2)} + P_{MD}^{(1)} P_{MD}^{(2)} < 1 + P_{FA}^{(1)} P_{FA}^{(2)}$

$P(T|T_o, T_o) > p \quad iff \quad P_{MD}^{(1)} + P_{MD}^{(2)} + P_{FA}^{(1)} P_{FA}^{(2)} < 1 + P_{MD}^{(1)} P_{MD}^{(2)}$

A reasonable choice is $P_t = p$, in particular if the problem parameters $P_{MD}^{(1)}, P_{MD}^{(2)}, P_{FA}^{(1)}, P_{FA}^{(2)}$ are s.t. $P(T|T_o, FT_o) < p < P(T|FT_o, T_o)$.

Consider now the choice

$$P(T|FT_o, FT_o) < P(T|T_o, FT_o) < P_t < P(T|FT_o, T_o) < P(T|T_o, T_o).$$

A classification error will occur if

1. The inspected object is a T but the operator said (FT_o, FT_o), or the operator said (T_o, FT_o) and the object will be misclassified as a FT

or,

2. The inspected object is a FT but the operator said (FT_o, T_o) or the operator said (T_o, T_o) and the object will be misclassified a T.

So

$$P_{E_1} = P_{MD}^{(1)}P_{MD}^{(2)} + P_{TR}^{(1)}P_{MD}^{(2)} = P_{MD}^{(2)}$$

$$P_{E_2} = P_{FTR}^{(1)}P_{FA}^{(2)} + P_{FA}^{(1)}P_{FA}^{(2)} = P_{FA}^{(2)}$$

Hence

$$P_E = pP_{E_1} + (1-p)P_{E_2}$$

=>

$$P_E = pP_{MD}^{(2)} + (1-p)P_{FA}^{(2)}$$

Appendix I: CCA Output File

I.1 Output File Format

The result of a run of 4 UAVs viewing 20 objects is a .csv (Excel) file containing a table of 37 columns and 96 rows. Each row is a report entry. As shown in the first column below, an entry of type 1 is the 1st look at an object, also known as a Point of Interest (POI). A type 2 entry is a revisit to a POI. Entry type 3 essentially records when the alert is cued. Entry type 4 records when the object enters the footprint. This could be used to ensure that the video is in sync with the data and to compute how long the operator took to respond to the alert. Column 2 is the simulation time the event took place, which goes from 0 seconds to about 700 seconds, or a run time of about 12 minutes. Column 3, the vehicle ID, is a 9 digit number. The last digit is [1,2,3,4]; since the file entries are in chronological order, this allows all the entries corresponding to a particular UAV to be grouped together. Columns 4,5 is the UAV heading and altitude. Column 6 is the POI or object number from 1-20. If the ID is a 1000 number, then the object is a true target (Y). Column 7-10 is the object 3Dim position and orientation (pose angle). Column 11,12 is the planner commanded UAV aspect angle and altitude for a type 1 entry, and the SC commanded values for a type 2 entry (revisit). Columns 13-16 are the actual values achieved. This shows how well the autopilot is tracking the commanded values. These 4 values constitute the true state that is used to interrogate the operator error model (table). Column 17,18 is the operator error model output values (Ptr=1-Pmd,Pftr=1-Pfa) at the error model input state (column 13-16). These are conditional probabilities (P(Op=T|T) and P(Op=NT|NT)). Columns 19,20 are the posterior calculations from Bayes, given the prior (column 22), the conditionals (column 17,18) and the operator observation (column 25). Column 19 is for 1 observation and column 20 is for 2 operator observations. Column 21 should agree with POI. Columns 23-33 are for the revisit type 2 data entry. Column 23 tells which revisit [1,2.3] the entry is for. Column

24 is the number of revisits remaining [3,2,1,0]. Column 26 is the output of the Stochastic Controller where the decision is made whether not to revisit [0], or if to revisit, at which state [1-256]. The state is the row number in the operator error model. Columns 27-33 are the commanded and actual revisit states. Only AA and GSD are outputs from the SC. Columns 34-36 are for the pseudo speed control. While Dwell Time is not commanded (or controlled) by the SC, the revisit speed is commanded by the planner to be the longest feasible Dwell Time (slowest speed). The IET is not commanded or controlled. Column 37 is the operator stated confidence level [1-5] in his classification decision (T/NT).

1. Type: [1=first response, 2=revisit response, 3=cue triggered, 4=entered footprint]
2. Time: [0-700] seconds
3. Vehicle ID: [xxxxxxxxx] (last digit 1-4)
4. Vehicle Heading(deg): [0-360] degrees, clockwise from N
5. Vehicle Altitude MSL(ft): [200-300] feet
6. POI ID: [1-20] (1000+ for target)
7. POI Latitude(deg): approximately 33 deg
8. POI Longitude(deg): approximately 44 deg
9. POI Altitude MSL(ft): [100] ft nominal
10. POI Heading(deg): [0-360] degrees, clockwise from N
11. Aspect Angle desired(deg) [45-90] degrees, from nose clockwise:
12. GSD desired(ft): [.2-.3] feet/pixel
13. AA actual(deg): [0-180] deg
14. GSD actual(ft): [.2-.3] feet/pixel
15. Dwell Time actual(sec): [2-5] sec
16. IET actual(sec): [0-120] sec
17. PTR: Probability of Target Report (0-1)
18. PFTR: Probability of False Target Report (0-1)

19. PT1: Prob of being a Target given report 1 (0-1)
20. PT2: Prob of being a Target given report 2 (0-1)
21. Target Truth: [0=NonTarget, 1=Target],(only shows up on reports 1 and 2)
22. p: the a priori density [.5-.05]
23. Revisit Number: [1,2,3]
24. Revisits Left: [3,2,1,0]
25. Operator Response: [1 = target, 0 = no target]
26. Revisit Decision: [0 = no revisit, >0 indicates desired revisit state]
27. Revisit AA commanded(deg): [90] nominally
28. Revisit GSD commanded(ft): [.19] ft/pixel nominally
29. Revisit Altitude Commanded MSL(ft): [200] nominally
30. Revisit AA actual: [90] deg nominal
31. Revisit GSD actual: [.19] nominal
32. Revisit Dwell Time actual: [4.5] sec nominal
33. Revisit IET actual: [0-100] sec
34. Commanded Dwell Time: [4.5] sec nominal
35. Commanded Speed: [29-60] fps, on revisit 29fps nominal
36. Speed actual: [29-60] fps, on revisit 29fps nominal
37. Operator Confidence Level: [1–5], 1 low, 5 high

This example output file is `20130909_3_2.csv`. It was taken on 9 September, 2013. It was one of 5 runs that were made that day by Operator S for the operator error model. It is Scenario 3 and it is the 2nd time scenario 3 was run that day.

Due to the size of this 97row by 37col table, it is broken up into 12 pages for readability. Pages 1-3 contain all the rows for the 1st 10 columns, pages 4-6 the 2nd 10 columns, pages 7-9 the 3rd 10 columns, and pages 10-12 the final 6 columns.

On page 3 is the summary results for this run. The operator had 2 FA, 2 MD, and no revisit errors.

The respective columns are:

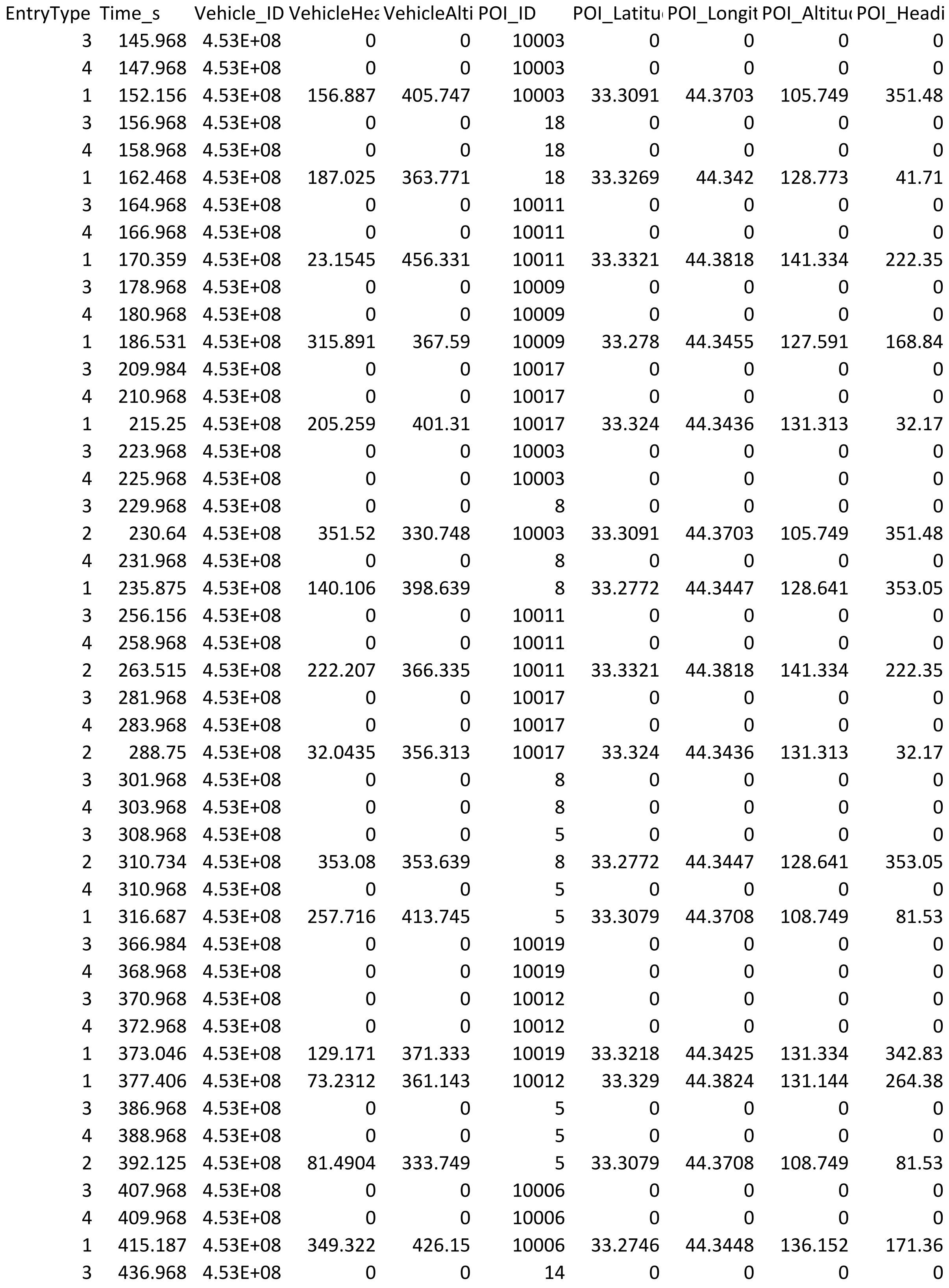
1. Type: [1=first response, 2=revisit response, 3=cue triggered, 4=entered footprint]; 2. Time: [0-700] seconds; 3. Vehicle ID: [xxxxxxxxx] (last digit 1-4); 4. Vehicle Heading(deg): [0-360] degrees, clockwise from N; 5. Vehicle Altitude MSL(ft): [200-300] feet; 6. POI ID: [1-20] (1000+ for target); 7. POI Latitude(deg): approximately 33 deg; 8. POI Longitude(deg): approximately 44 deg; 9. POI Altitude MSL(ft): [100] ft nominal; 10. POI Heading(deg): [0-360] degrees, clockwise from N

11. Aspect Angle desired(deg) [45-90] degrees, from nose clockwise; 12. GSD desired(ft): [.2-.3] feet/pixel; 13. AA actual(deg): [0-180] deg; 14. GSD actual(ft): [.2-.3] feet/pixel; 15. Dwell Time actual(sec): [2-5] sec; 16. IET actual(sec): [0-120] sec; 17. PTR: Probability of Target Report (0-1); 18. PFTR: Probability of False Target Report (0-1); 19. PT1: Prob of being a Target given report 1 (0-1); 20. PT2: Prob of being a Target given report 2 (0-1)

21. Target Truth: [0=NonTarget, 1=Target],(only shows up on reports 1 and 2); 22. p: the a priori density [.5-.05]; 23. Revisit Number: [1,2,3]; 24. Revisits Left: [3,2,1,0]; 25. Operator Response: [1 = target, 0 = no target]; 26. Revisit Decision: [0 = no revisit, >0 indicates desired revisit state]; 27. Revisit AA commanded(deg): [90] nominally; 28. Revisit GSD commanded(ft): [.19] ft/pixel nominally; 29. Revisit Altitude Commanded MSL(ft): [200] nominally; 30. Revisit AA actual: [90] deg nominal

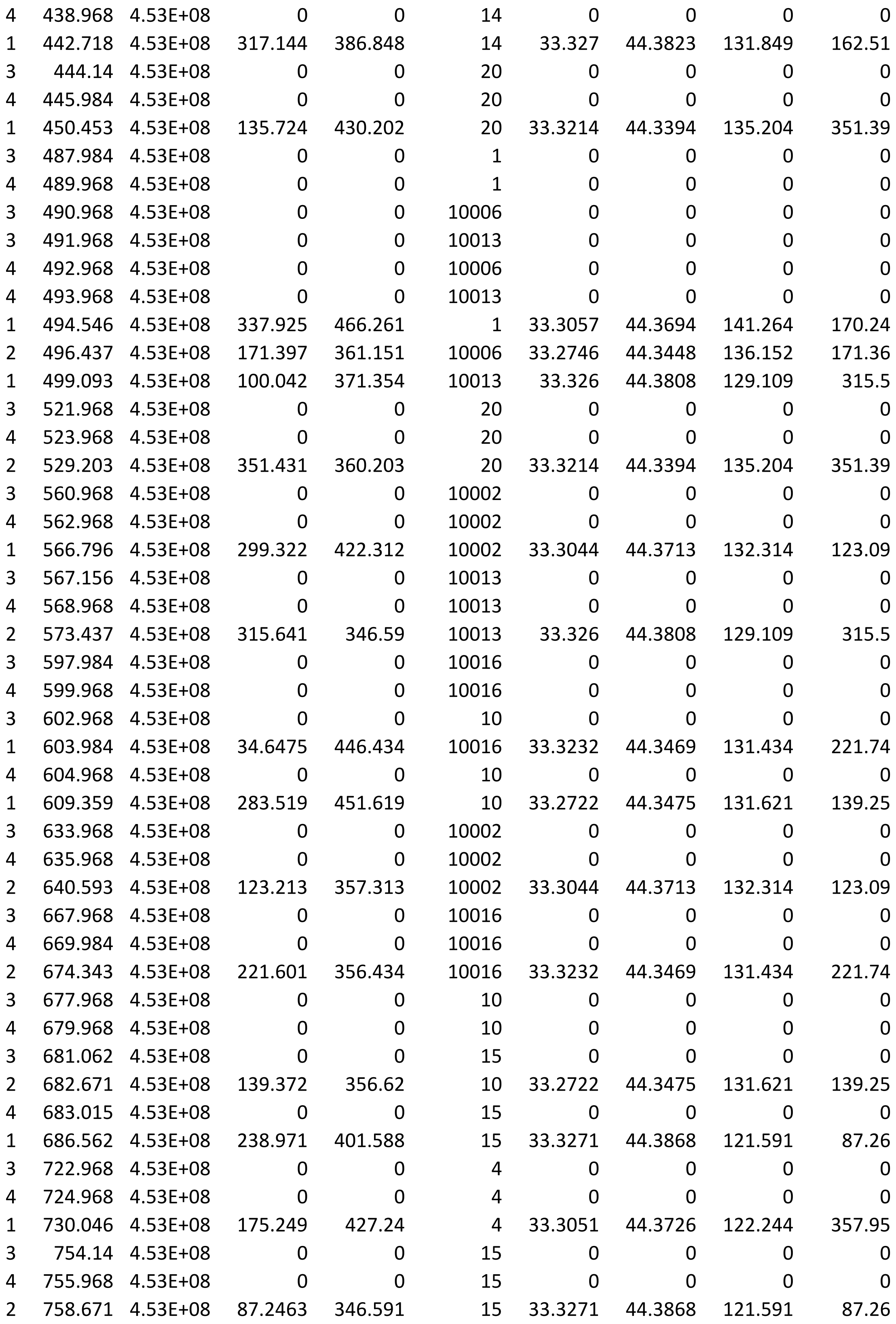
31. Revisit GSD actual: [.19] nominal; 32. Revisit Dwell Time actual: [4.5] sec nominal; 33. Revisit IET actual: [0-100] sec; 34. Commanded Dwell Time: [4.5] sec nominal;

35. Commanded Speed: [29-60] fps, on revisit 29fps nominal; 36. Speed actual: [29-60] fps, on revisit 29fps nominal



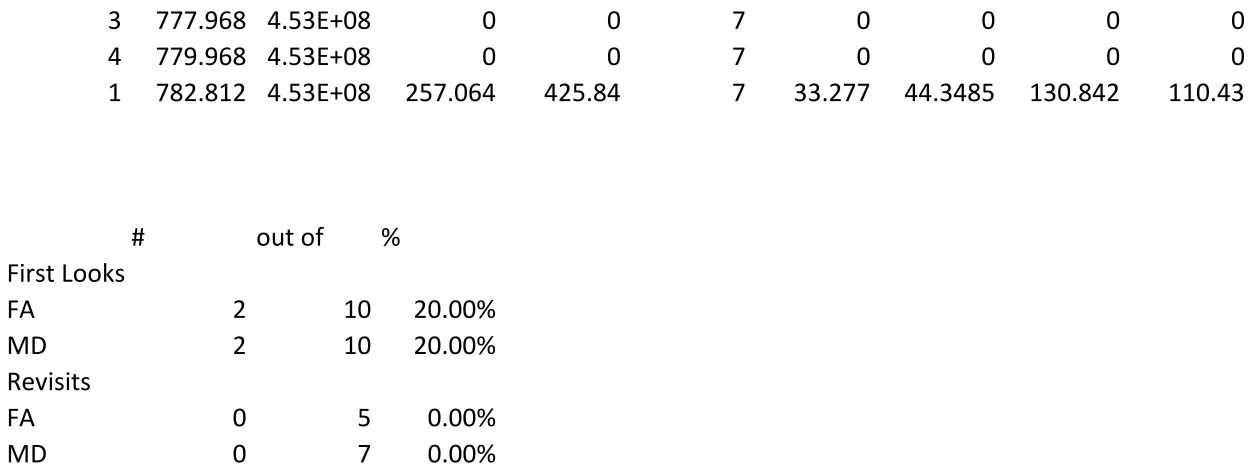
EntryType	Time_s	Vehicle_ID	VehicleHea	VehicleAlti	POI_ID	POI_Latitu	POI_Longit	POI_Altitu	POI_Headi
3	145.968	4.53E+08	0	0	10003	0	0	0	0
4	147.968	4.53E+08	0	0	10003	0	0	0	0
1	152.156	4.53E+08	156.887	405.747	10003	33.3091	44.3703	105.749	351.48
3	156.968	4.53E+08	0	0	18	0	0	0	0
4	158.968	4.53E+08	0	0	18	0	0	0	0
1	162.468	4.53E+08	187.025	363.771	18	33.3269	44.342	128.773	41.71
3	164.968	4.53E+08	0	0	10011	0	0	0	0
4	166.968	4.53E+08	0	0	10011	0	0	0	0
1	170.359	4.53E+08	23.1545	456.331	10011	33.3321	44.3818	141.334	222.35
3	178.968	4.53E+08	0	0	10009	0	0	0	0
4	180.968	4.53E+08	0	0	10009	0	0	0	0
1	186.531	4.53E+08	315.891	367.59	10009	33.278	44.3455	127.591	168.84
3	209.984	4.53E+08	0	0	10017	0	0	0	0
4	210.968	4.53E+08	0	0	10017	0	0	0	0
1	215.25	4.53E+08	205.259	401.31	10017	33.324	44.3436	131.313	32.17
3	223.968	4.53E+08	0	0	10003	0	0	0	0
4	225.968	4.53E+08	0	0	10003	0	0	0	0
3	229.968	4.53E+08	0	0	8	0	0	0	0
2	230.64	4.53E+08	351.52	330.748	10003	33.3091	44.3703	105.749	351.48
4	231.968	4.53E+08	0	0	8	0	0	0	0
1	235.875	4.53E+08	140.106	398.639	8	33.2772	44.3447	128.641	353.05
3	256.156	4.53E+08	0	0	10011	0	0	0	0
4	258.968	4.53E+08	0	0	10011	0	0	0	0
2	263.515	4.53E+08	222.207	366.335	10011	33.3321	44.3818	141.334	222.35
3	281.968	4.53E+08	0	0	10017	0	0	0	0
4	283.968	4.53E+08	0	0	10017	0	0	0	0
2	288.75	4.53E+08	32.0435	356.313	10017	33.324	44.3436	131.313	32.17
3	301.968	4.53E+08	0	0	8	0	0	0	0
4	303.968	4.53E+08	0	0	8	0	0	0	0
3	308.968	4.53E+08	0	0	5	0	0	0	0
2	310.734	4.53E+08	353.08	353.639	8	33.2772	44.3447	128.641	353.05
4	310.968	4.53E+08	0	0	5	0	0	0	0
1	316.687	4.53E+08	257.716	413.745	5	33.3079	44.3708	108.749	81.53
3	366.984	4.53E+08	0	0	10019	0	0	0	0
4	368.968	4.53E+08	0	0	10019	0	0	0	0
3	370.968	4.53E+08	0	0	10012	0	0	0	0
4	372.968	4.53E+08	0	0	10012	0	0	0	0
1	373.046	4.53E+08	129.171	371.333	10019	33.3218	44.3425	131.334	342.83
1	377.406	4.53E+08	73.2312	361.143	10012	33.329	44.3824	131.144	264.38
3	386.968	4.53E+08	0	0	5	0	0	0	0
4	388.968	4.53E+08	0	0	5	0	0	0	0
2	392.125	4.53E+08	81.4904	333.749	5	33.3079	44.3708	108.749	81.53
3	407.968	4.53E+08	0	0	10006	0	0	0	0
4	409.968	4.53E+08	0	0	10006	0	0	0	0
1	415.187	4.53E+08	349.322	426.15	10006	33.2746	44.3448	136.152	171.36
3	436.968	4.53E+08	0	0	14	0	0	0	0

Figure I.1: Output File, Row 1–46, Col 1–10



4	438.968	4.53E+08	0	0	14	0	0	0	0
1	442.718	4.53E+08	317.144	386.848	14	33.327	44.3823	131.849	162.51
3	444.14	4.53E+08	0	0	20	0	0	0	0
4	445.984	4.53E+08	0	0	20	0	0	0	0
1	450.453	4.53E+08	135.724	430.202	20	33.3214	44.3394	135.204	351.39
3	487.984	4.53E+08	0	0	1	0	0	0	0
4	489.968	4.53E+08	0	0	1	0	0	0	0
3	490.968	4.53E+08	0	0	10006	0	0	0	0
3	491.968	4.53E+08	0	0	10013	0	0	0	0
4	492.968	4.53E+08	0	0	10006	0	0	0	0
4	493.968	4.53E+08	0	0	10013	0	0	0	0
1	494.546	4.53E+08	337.925	466.261	1	33.3057	44.3694	141.264	170.24
2	496.437	4.53E+08	171.397	361.151	10006	33.2746	44.3448	136.152	171.36
1	499.093	4.53E+08	100.042	371.354	10013	33.326	44.3808	129.109	315.5
3	521.968	4.53E+08	0	0	20	0	0	0	0
4	523.968	4.53E+08	0	0	20	0	0	0	0
2	529.203	4.53E+08	351.431	360.203	20	33.3214	44.3394	135.204	351.39
3	560.968	4.53E+08	0	0	10002	0	0	0	0
4	562.968	4.53E+08	0	0	10002	0	0	0	0
1	566.796	4.53E+08	299.322	422.312	10002	33.3044	44.3713	132.314	123.09
3	567.156	4.53E+08	0	0	10013	0	0	0	0
4	568.968	4.53E+08	0	0	10013	0	0	0	0
2	573.437	4.53E+08	315.641	346.59	10013	33.326	44.3808	129.109	315.5
3	597.984	4.53E+08	0	0	10016	0	0	0	0
4	599.968	4.53E+08	0	0	10016	0	0	0	0
3	602.968	4.53E+08	0	0	10	0	0	0	0
1	603.984	4.53E+08	34.6475	446.434	10016	33.3232	44.3469	131.434	221.74
4	604.968	4.53E+08	0	0	10	0	0	0	0
1	609.359	4.53E+08	283.519	451.619	10	33.2722	44.3475	131.621	139.25
3	633.968	4.53E+08	0	0	10002	0	0	0	0
4	635.968	4.53E+08	0	0	10002	0	0	0	0
2	640.593	4.53E+08	123.213	357.313	10002	33.3044	44.3713	132.314	123.09
3	667.968	4.53E+08	0	0	10016	0	0	0	0
4	669.984	4.53E+08	0	0	10016	0	0	0	0
2	674.343	4.53E+08	221.601	356.434	10016	33.3232	44.3469	131.434	221.74
3	677.968	4.53E+08	0	0	10	0	0	0	0
4	679.968	4.53E+08	0	0	10	0	0	0	0
3	681.062	4.53E+08	0	0	15	0	0	0	0
2	682.671	4.53E+08	139.372	356.62	10	33.2722	44.3475	131.621	139.25
4	683.015	4.53E+08	0	0	15	0	0	0	0
1	686.562	4.53E+08	238.971	401.588	15	33.3271	44.3868	121.591	87.26
3	722.968	4.53E+08	0	0	4	0	0	0	0
4	724.968	4.53E+08	0	0	4	0	0	0	0
1	730.046	4.53E+08	175.249	427.24	4	33.3051	44.3726	122.244	357.95
3	754.14	4.53E+08	0	0	15	0	0	0	0
4	755.968	4.53E+08	0	0	15	0	0	0	0
2	758.671	4.53E+08	87.2463	346.591	15	33.3271	44.3868	121.591	87.26

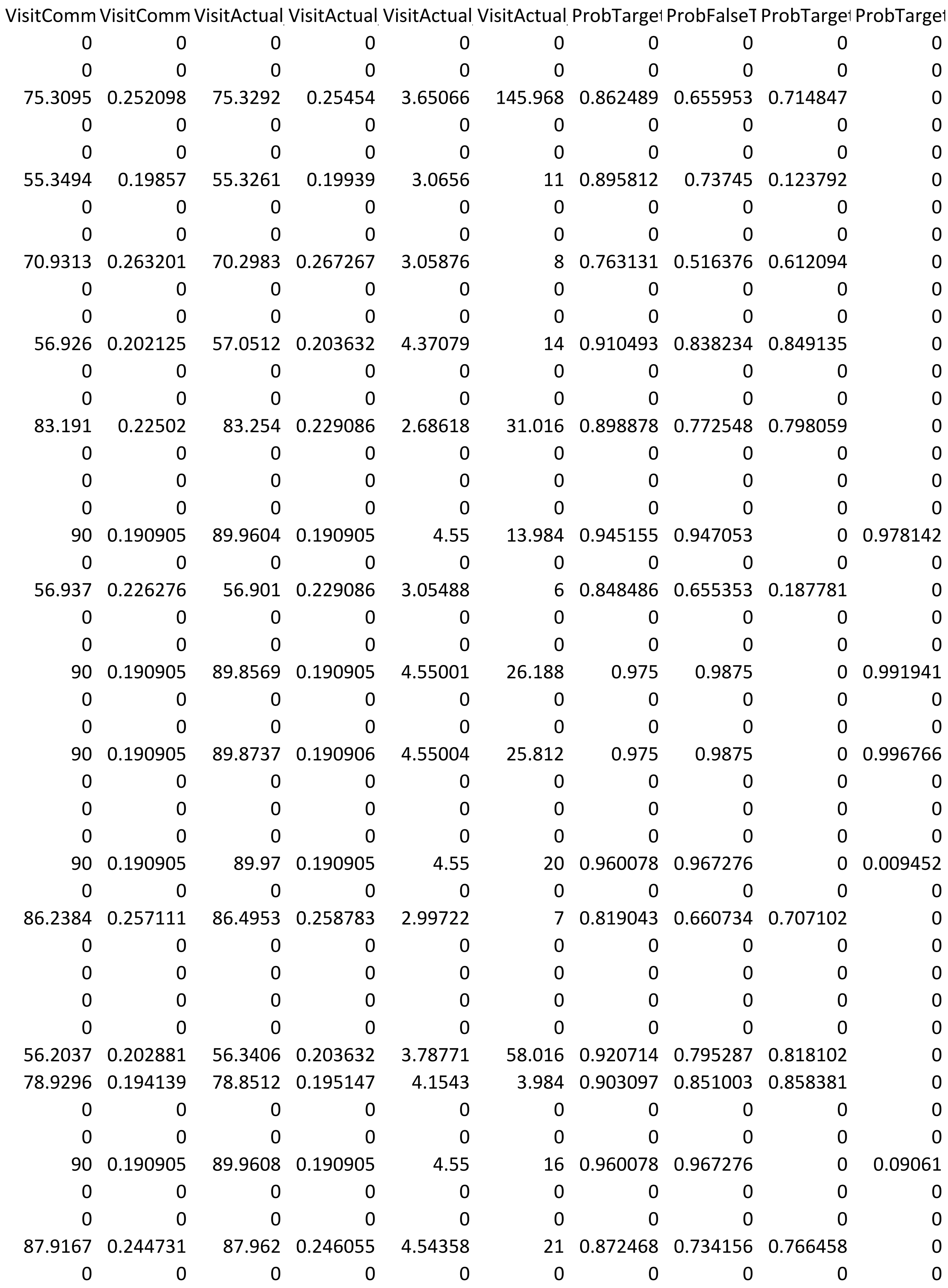
Figure I.2: Output File, Row 47–94, Col 1–10



3	777.968	4.53E+08	0	0		7	0	0	0	0
4	779.968	4.53E+08	0	0		7	0	0	0	0
1	782.812	4.53E+08	257.064	425.84		7	33.277	44.3485	130.842	110.43

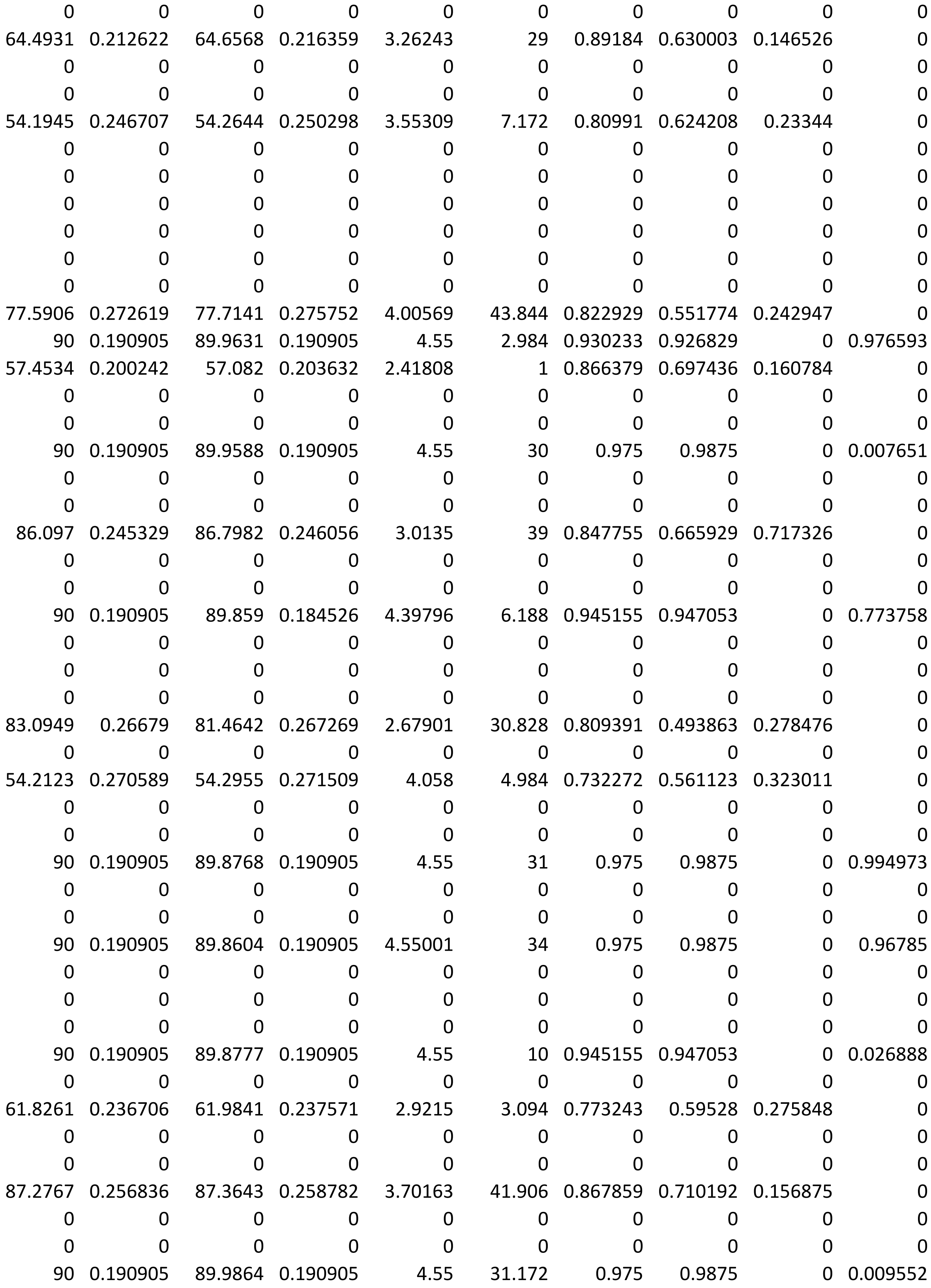
	#	out of	%
First Looks			
FA	2	10	20.00%
MD	2	10	20.00%
Revisits			
FA	0	5	0.00%
MD	0	7	0.00%

Figure I.3: Output File, Row 95–97, Col 1–10



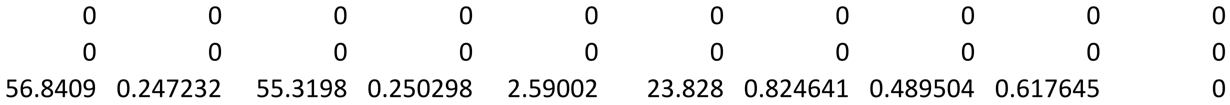
VisitComm	VisitComm	VisitActual	VisitActual	VisitActual	VisitActual	ProbTarget	ProbFalseT	ProbTarget	ProbTarget
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
75.3095	0.252098	75.3292	0.25454	3.65066	145.968	0.862489	0.655953	0.714847	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
55.3494	0.19857	55.3261	0.19939	3.0656	11	0.895812	0.73745	0.123792	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
70.9313	0.263201	70.2983	0.267267	3.05876	8	0.763131	0.516376	0.612094	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
56.926	0.202125	57.0512	0.203632	4.37079	14	0.910493	0.838234	0.849135	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
83.191	0.22502	83.254	0.229086	2.68618	31.016	0.898878	0.772548	0.798059	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
90	0.190905	89.9604	0.190905	4.55	13.984	0.945155	0.947053	0	0.978142
0	0	0	0	0	0	0	0	0	0
56.937	0.226276	56.901	0.229086	3.05488	6	0.848486	0.655353	0.187781	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
90	0.190905	89.8569	0.190905	4.55001	26.188	0.975	0.9875	0	0.991941
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
90	0.190905	89.8737	0.190906	4.55004	25.812	0.975	0.9875	0	0.996766
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
90	0.190905	89.97	0.190905	4.55	20	0.960078	0.967276	0	0.009452
0	0	0	0	0	0	0	0	0	0
86.2384	0.257111	86.4953	0.258783	2.99722	7	0.819043	0.660734	0.707102	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
56.2037	0.202881	56.3406	0.203632	3.78771	58.016	0.920714	0.795287	0.818102	0
78.9296	0.194139	78.8512	0.195147	4.1543	3.984	0.903097	0.851003	0.858381	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
90	0.190905	89.9608	0.190905	4.55	16	0.960078	0.967276	0	0.09061
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
87.9167	0.244731	87.962	0.246055	4.54358	21	0.872468	0.734156	0.766458	0
0	0	0	0	0	0	0	0	0	0

Figure I.4: Output File, Row 1–46, Col 11–20



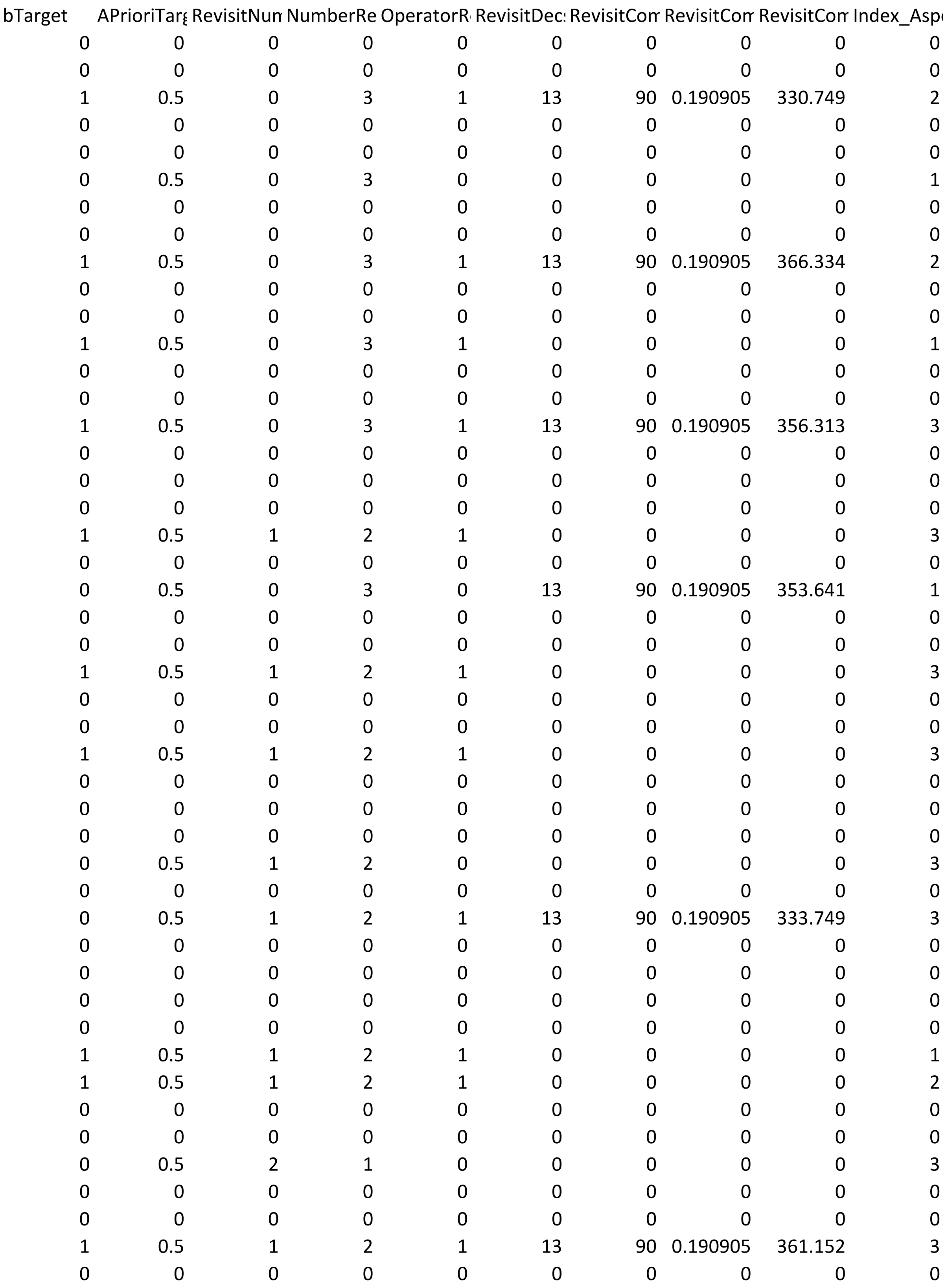
```
         0           0           0           0           0           0           0           0           0           0
   64.4931    0.212622     64.6568    0.216359     3.26243          29     0.89184    0.630003    0.146526           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
   54.1945    0.246707     54.2644    0.250298     3.55309       7.172     0.80991    0.624208     0.23344           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
   77.5906    0.272619     77.7141    0.275752     4.00569      43.844    0.822929    0.551774    0.242947           0
        90    0.190905     89.9631    0.190905        4.55       2.984    0.930233    0.926829           0    0.976593
   57.4534    0.200242      57.082    0.203632     2.41808           1    0.866379    0.697436    0.160784           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
        90    0.190905     89.9588    0.190905        4.55          30       0.975      0.9875           0    0.007651
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
    86.097    0.245329     86.7982    0.246056      3.0135          39    0.847755    0.665929    0.717326           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
        90    0.190905      89.859    0.184526     4.39796       6.188    0.945155    0.947053           0    0.773758
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
   83.0949     0.26679     81.4642    0.267269     2.67901      30.828    0.809391    0.493863    0.278476           0
         0           0           0           0           0           0           0           0           0           0
   54.2123    0.270589     54.2955    0.271509       4.058       4.984    0.732272    0.561123    0.323011           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
        90    0.190905     89.8768    0.190905        4.55          31       0.975      0.9875           0    0.994973
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
        90    0.190905     89.8604    0.190905     4.55001          34       0.975      0.9875           0     0.96785
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
        90    0.190905     89.8777    0.190905        4.55          10    0.945155    0.947053           0    0.026888
         0           0           0           0           0           0           0           0           0           0
   61.8261    0.236706     61.9841    0.237571      2.9215       3.094    0.773243     0.59528    0.275848           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
   87.2767    0.256836     87.3643    0.258782     3.70163      41.906    0.867859    0.710192    0.156875           0
         0           0           0           0           0           0           0           0           0           0
         0           0           0           0           0           0           0           0           0           0
        90    0.190905     89.9864    0.190905        4.55      31.172       0.975      0.9875           0    0.009552
```

Figure I.5: Output File, Row 47–94, Col 11–20



```
       0         0         0         0         0         0         0         0         0         0
       0         0         0         0         0         0         0         0         0         0
 56.8409  0.247232   55.3198  0.250298   2.59002    23.828  0.824641  0.489504  0.617645         0
```

Figure I.6: Output File, Row 95–97, Col 11–20



bTarget	APrioriTarg	RevisitNum	NumberRe	OperatorR	RevisitDec	RevisitCom	RevisitCom	RevisitCom	Index_Asp
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0.5	0	3	1	13	90	0.190905	330.749	2
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0.5	0	3	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0.5	0	3	1	13	90	0.190905	366.334	2
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0.5	0	3	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0.5	0	3	1	13	90	0.190905	356.313	3
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0.5	1	2	1	0	0	0	0	3
0	0	0	0	0	0	0	0	0	0
0	0.5	0	3	0	13	90	0.190905	353.641	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0.5	1	2	1	0	0	0	0	3
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0.5	1	2	1	0	0	0	0	3
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0.5	1	2	0	0	0	0	0	3
0	0	0	0	0	0	0	0	0	0
0	0.5	1	2	1	13	90	0.190905	333.749	3
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0.5	1	2	1	0	0	0	0	1
1	0.5	1	2	1	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0.5	2	1	0	0	0	0	0	3
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0.5	1	2	1	13	90	0.190905	361.152	3
0	0	0	0	0	0	0	0	0	0

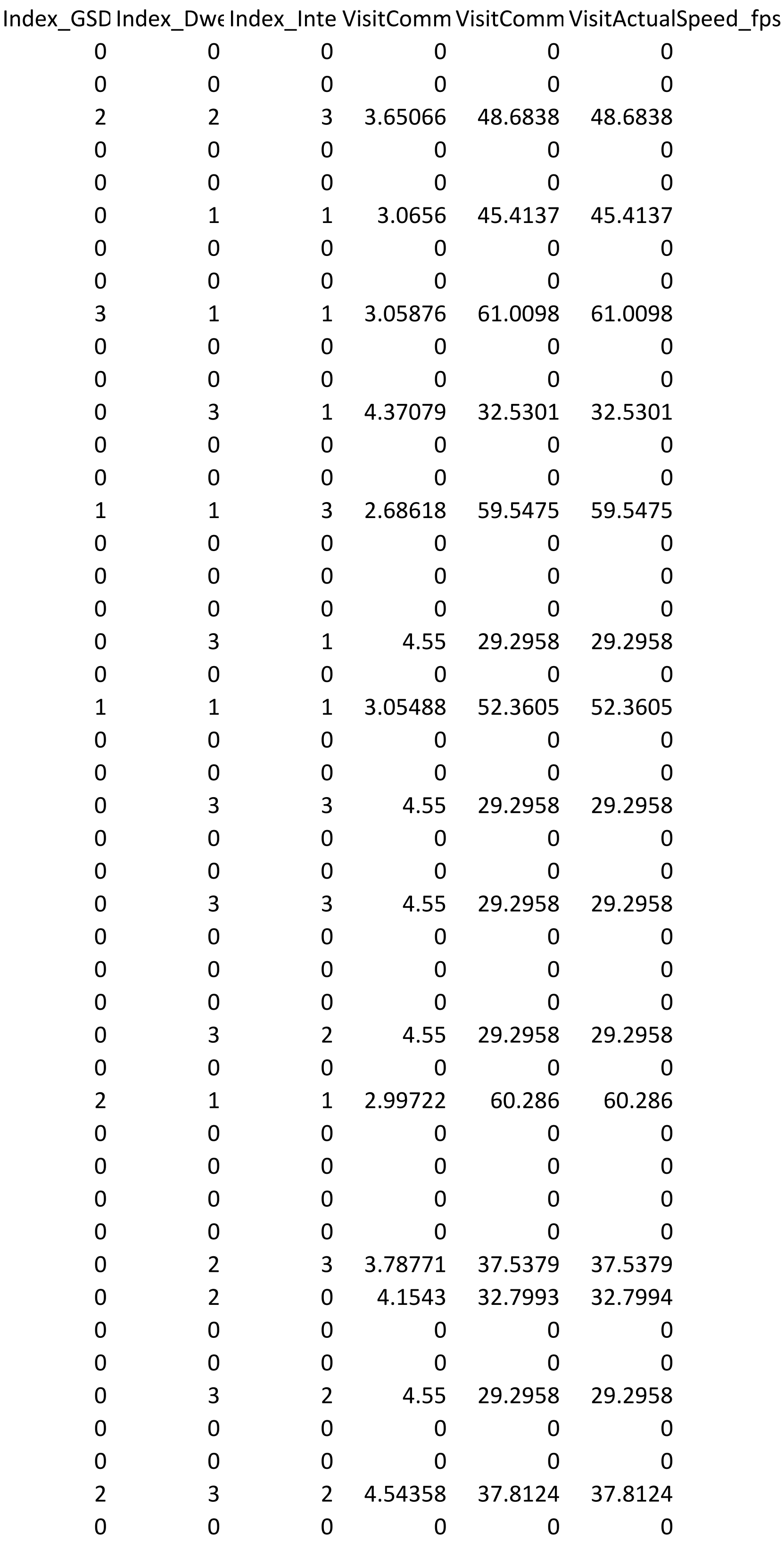
Figure I.7: Output File, Row 1–46, Col 21–30

```
0          0          0          0          0          0          0          0          0          0
0        0.5          1          2          0          0          0          0          0          1
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0        0.5          1          2          0         13         90   0.190905    360.204          1
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0        0.5          2          1          0          0          0          0          0          2
1        0.5          2          1          1          0          0          0          0          3
1        0.5          1          2          0         13         90   0.190905    346.591          1
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0        0.5          2          1          0          0          0          0          0          3
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
1        0.5          2          1          1         13         90   0.190905    357.313          3
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
1        0.5          2          1          1          0          0          0          0          3
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
1        0.5          2          1          0         13         90   0.190905    356.433          2
0          0          0          0          0          0          0          0          0          0
0        0.5          2          1          0         13         90   0.190905    356.621          1
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
1        0.5          3          0          1          0          0          0          0          3
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
1        0.5          3          0          1          0          0          0          0          3
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0        0.5          3          0          0          0          0          0          0          3
0          0          0          0          0          0          0          0          0          0
0        0.5          2          1          0         13         90   0.190905    346.591          1
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0        0.5          3          0          0          0          0          0          0          3
0          0          0          0          0          0          0          0          0          0
0          0          0          0          0          0          0          0          0          0
0        0.5          3          0          0          0          0          0          0          3
```

Figure I.8: Output File, Row 47–94, Col 21–30

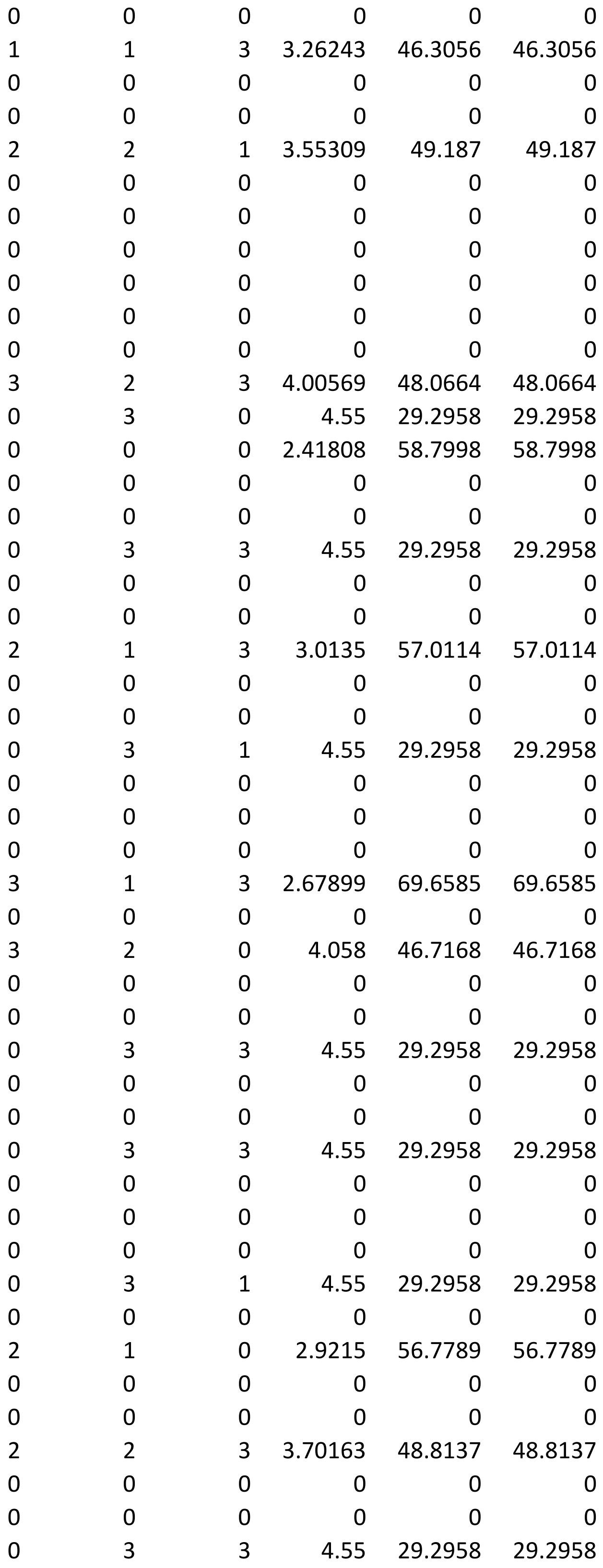
```
0      0      0      0      0      0      0      0      0      0
0      0      0      0      0      0      0      0      0      0
0    0.5      3      0      1      0      0      0      0      1
```

Figure I.9: Output File, Row 95–97, Col 21–30



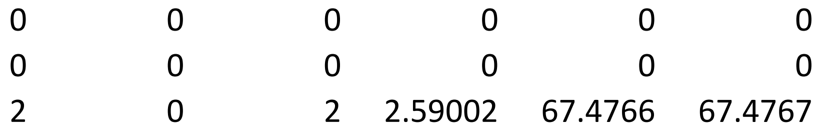
Index_GSD	Index_Dwe	Index_Inte	VisitComm	VisitComm	VisitActualSpeed_fps
0	0	0	0	0	0
0	0	0	0	0	0
2	2	3	3.65066	48.6838	48.6838
0	0	0	0	0	0
0	0	0	0	0	0
0	1	1	3.0656	45.4137	45.4137
0	0	0	0	0	0
0	0	0	0	0	0
3	1	1	3.05876	61.0098	61.0098
0	0	0	0	0	0
0	0	0	0	0	0
0	3	1	4.37079	32.5301	32.5301
0	0	0	0	0	0
0	0	0	0	0	0
1	1	3	2.68618	59.5475	59.5475
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	3	1	4.55	29.2958	29.2958
0	0	0	0	0	0
1	1	1	3.05488	52.3605	52.3605
0	0	0	0	0	0
0	0	0	0	0	0
0	3	3	4.55	29.2958	29.2958
0	0	0	0	0	0
0	0	0	0	0	0
0	3	3	4.55	29.2958	29.2958
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	3	2	4.55	29.2958	29.2958
0	0	0	0	0	0
2	1	1	2.99722	60.286	60.286
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	2	3	3.78771	37.5379	37.5379
0	2	0	4.1543	32.7993	32.7994
0	0	0	0	0	0
0	0	0	0	0	0
0	3	2	4.55	29.2958	29.2958
0	0	0	0	0	0
0	0	0	0	0	0
2	3	2	4.54358	37.8124	37.8124
0	0	0	0	0	0

Figure I.10: Output File, Row 1–46, Col 31–36



```
0      0      0         0         0         0
1      1      3   3.26243   46.3056   46.3056
0      0      0         0         0         0
0      0      0         0         0         0
2      2      1   3.55309    49.187    49.187
0      0      0         0         0         0
0      0      0         0         0         0
0      0      0         0         0         0
0      0      0         0         0         0
0      0      0         0         0         0
0      0      0         0         0         0
3      2      3   4.00569   48.0664   48.0664
0      3      0      4.55   29.2958   29.2958
0      0      0   2.41808   58.7998   58.7998
0      0      0         0         0         0
0      0      0         0         0         0
0      3      3      4.55   29.2958   29.2958
0      0      0         0         0         0
0      0      0         0         0         0
2      1      3    3.0135   57.0114   57.0114
0      0      0         0         0         0
0      0      0         0         0         0
0      3      1      4.55   29.2958   29.2958
0      0      0         0         0         0
0      0      0         0         0         0
0      0      0         0         0         0
3      1      3   2.67899   69.6585   69.6585
0      0      0         0         0         0
3      2      0     4.058   46.7168   46.7168
0      0      0         0         0         0
0      0      0         0         0         0
0      3      3      4.55   29.2958   29.2958
0      0      0         0         0         0
0      0      0         0         0         0
0      3      3      4.55   29.2958   29.2958
0      0      0         0         0         0
0      0      0         0         0         0
0      0      0         0         0         0
0      3      1      4.55   29.2958   29.2958
0      0      0         0         0         0
2      1      0    2.9215   56.7789   56.7789
0      0      0         0         0         0
0      0      0         0         0         0
2      2      3   3.70163   48.8137   48.8137
0      0      0         0         0         0
0      0      0         0         0         0
0      3      3      4.55   29.2958   29.2958
```

Figure I.11: Output File, Row 47–94, Col 31–36



```
0         0         0         0         0         0
0         0         0         0         0         0
2         0         2   2.59002   67.4766   67.4767
```

Figure I.12: Output File, Row 95–97, Col 31–36

Appendix J: Operator Debrief

Human Behavior Modeling & Stochastic Control Project

Outbrief Questions for S1

11 Oct 2013

Q: You have not been provided any feedback. Do you think this has hurt your performance (or score)? (PC=75%, FA=38%, MD=12%)

A: Thinks that feedback would have caused more stress to perform better. Leans toward having no feedback. Thought saw more Y than V at the more oblique aspect angles. Did not consciously bias the results—did not lean toward FA or MD.

Q: Do you think you are doing better over time? (72% start to 75% end; mean PC=73.5%; std 3.5%)

A: Thought performance was improving more than what it actually did. Later in the runs, confidence scores went up.

Q: No criterion has been given to you—can you describe your decision criterion?

A: Said that if the aspect angle is straight on, then can render a decision in 1 sec. If can't tell, then will weight the T/NT decision with the number of targets seen over the last 4-5 objects. If there was a state that knew wouldn't be able to see well (e.g. high altitude and low angle) then spent less effort to identify those vehicles.

Q: After 130 runs, do you think you can predict what comes next? Do you recognize individual scenarios? How many are there?

A: Knows there are 4 scenarios and can recognize them as such. However, cannot predict if the next object is going to be a T or NT.

Q: What do you think the density is? How do you think this affects your criteria?

A: Was told at the beginning of the study that the density was 50%. Recognized sometimes that the same area was being looked at again, but wasn't aware that it was a second look.

Stated that what was said on the 2nd look was not affected by what was said on the 1st look—independent observations.

Q: Which conditions (or states) do you think are hardest for you and which are easiest?

A: Said the highest altitude and the most oblique angle is essentially a guess. Said Dwell and IET variations are much less problematic.

Q: Do you think the erratic video has appreciably affected your score?

A: The video glitch would occasionally cause an object to be missed on multiple alerts. Rarely would the glitch "help".

Q: Have you noticed any situations where all 4 alerts are on?

A: There would be 4 alerts on about twice per session (5 runs). Of these, would generally guess on 1 of the 4.

Q: How often do you see 3 alerts? 2 alerts? Degree of difficulty? Do you guess sometimes with 3 alerts? Miss? (miss means did not have time to observe object and did not select T/NT)

A: Said had a 3 alert event in every scenario. Did not "miss" any of the objects.

Q: If there were no more than 1 alert at a time, do you think this would appreciably improve your score?

A: Said 1 at time probably would not be much better. Can easily handle 2 at time, even 3.

Q: Do you think the Y-V discrimination task is too contrived? (or trivial?)

A: Said, on average, the discrimination task is moderately difficult. At higher depression angles the V looks like a Y.

Q: What would you change about the simulation or task to make it more "realistic".

A: Have people and vehicles moving around.

Q: Are the sessions boring? Should they be shorter or longer? Fatigue? Stress? Eye strain? (Note: we are trying to estimate distributions, which take a lot more data—use aero table analogy.)

A: Boring—yes. But doesn't think zoned out, because each run and session are sufficiently short.

Q: If you had 3 choices - (T, NT, ["Don't Know" V "Can't Tell" V "Guess" V "Not Sure" V "Ambiguous" V "Equally Likely"]), how often do you think you would have picked the 3rd choice?

A: Felt that if had had a 3rd choice from the beginning, would have abused it (used it most of the time). Now, thinks could use it judiciously.

Q: How do you interpret a 1–5 confidence ranking? If you had to map confidence ranking into the 3rd choice button, which would you choose? (1 and 2)?

A: Would choose 1 and 2. Test Director and operator said felt there was a change in the confidence levels on or about 4 Sep. (Note: This date seems to be correlated with the change in the Operator Model.) Felt there were more 1+2 before this date than after. (Note: This was checked and there did not seem to be a statistically significant difference.)

Q: Do you think the initial training you received was adequate? How would you change it?

A: Thought the training and duration (5 runs) was adequate. Would prefer to have seen a V before the training.

Q: Suppose you knew up front that the "score" metric was PC. Do you think that would change or bias the present results? What if you got continuous updates?

A: Felt that knowing about the PC metric probably would not have helped. Liked or preferred not knowing about the score or incremental performance. Knowing would have caused more second guessing.

(Explain how the SC works if the operator doesn't already know.)

Q: As you now know, you were performing the classifier task—everything else was automated. Would you have preferred more control over the task? The planner? More explanation? Tactical Situation Display (TSD)? Do you think this would have improved the final results? (Estimator PC=93.5%)

A: Said would have decided to revisit on a confidence level of 1+2.

Appendix K: Alternative Revisit Strategy Based on Confidence Ratings

The Stochastic Controller (SC) attempts to improve overall performance by selecting points of interest for revisit. It makes selections and attempts to do so in an optimal and tractable way. In an attempt to compare the SC's revisits strategy to an alternative revisit strategy, we explored the human's ability to select a revisit. Also, using the human to select revisits has the advantage of being able to generalize to different targets more easily than the stochastic controller and the human behavior model. Since the operator did not report her desire to revisit during the experiment, we approximated that decision using the confidence responses with the rationale that an operator would revisit when confidence is low (e.g., she knew she made a guess).

The revisit performance data was approximated by selecting low confidence ratings and replacing the initial visit score with the average score for revisits, 98.75%. To reiterate, when a point of interest was revisited, the response from the initial visit score (and any information contained within it) was lost. Low confidence ratings of 1 were prioritized over confidence ratings of 2. The confidence rating of 1 was given on approximately 37% of initial visits and the UAVs are able to revisit 60% points of interest in total, leaving 23% remaining for confidence ratings of 2. Since there were more confidence ratings of 2 than could be revisited, the remaining budget was spent on a random selection of confidence ratings of 2. The confidence-based revisit strategy resulted in an overall system performance of 92.86%. This score differs slightly from the overall system performance of 93.5% when the SC was used.

One should note that a rigorous approach to using the operator's confidence rating requires the expansion of the confusion matrix. Experimentally obtaining such an augmented confusion matrix would be expensive, particularly at low target density.

Appendix L: Simulation Setup Procedure

1. Power on VSCS Computer and log on
    (a) User: usaf
    (b) Pwd: xxxxx
2. Power on all MetaVR computers
    (a) User: vssensor
    (b) Pwd: xxxxxxxx
3. Open TrueCrypt and Mount D drive
    (a) Select encryption file location: C:\D\encypted.tc
    (b) Mount with pwd: xxxxxxxx
4. Move the scenario appropriate terrain file into each of the 4 networked UAV drives
    (a) Ensure that it is named ”Baghdad-Demo_Terrain.clt”
5. Launch MetaVR on each sensor computer
    (a) Double click VSSensor icon on desktop
    (b) Ensure pointer is set to correct network configuration
    (c) Hit green arrow
    (d) Once loading is done, open MetaVR console (press exit key on keyboard) and then hit record (ensure video is set to 640x480 @ 8 bits/sec)
6. Move the correct scenario mission file into the mission folder
    (a) PointSearchTasks.vsxml

7. Move the correct operator model into the model folder

   (a) operatorModel.xml

8. Start the Control station and simulator

   (a) Double click on the launcher

   (b) Ensure version is correct

   (c) Select both GCS and Sim and launch

   (d) Ensure mission and network configuration pointers are correct

   (e) Check ”Use Video Takers” on the simulation panel

   (f) Launch both GCS and Sim

9. Start the simulation (ensure that MetaVR sims are all running before doing this)

10. In the assets tab of the bottom pane, select ”Full control” on each of the UAVs

11. Configure TSD

    (a) Map-> CB 10m

    (b) Center on -> Hawk A, Center on-> none

    (c) General Declutter -> uncheck UAV video/info and anything else

12. Configure video feeds

    (a) Starting from top left = Hawk A, top right = Hawk B, bottom left = Hawk C, bottom right = Hawk D

    (b) Deselect HUD on all UAVs

13. Start CCA

    (a) Click on CCA Tab

(b) Start planning

(c) Next

(d) Select all UAVs

(e) Next

(f) Submit plan request

(g) Check auto upload plans

(h) Once plans flash green click ”All on”. Will ask if you are sure. Click ”yes” to begin.

14. To start a new scenario, close VSCS , Simulation, and each of the MetaVR sensors. Wait for GCS to fully close and ensure any extra command windows are closed. Start over from step 4.